



# Movicon.next<sup>TM</sup>

## Help Manual

English Ed.– Rel. 4.1.333



# Table of Contents

<b>1. 1. INTRODUCTION .....</b>	<b>1</b>
1.1. MOVICON.NEXT & WEBHMI .....	1
1.2. PROJECT ARCHITECTURE .....	2
1.3. IMPORTANT INFORMATION .....	3
1.3.1. Preface.....	3
1.3.2. Warranty and Responsibility.....	4
1.3.3. Copyright.....	5
1.3.4. EndUser License Agreement (EULA).....	5
1.3.5. Open Source Components.....	10
1.3.6. Software Product Lifecycle Policy.....	15
1.3.7. Waste Disposal.....	18
 <b>2. THE DEVELOPMENT ENVIRONMENT .....</b>	 <b>21</b>
2.1. WORKSPACE .....	21
2.2. STARTUP PAGE .....	24
2.3. MENUS AND TOOLBARS .....	28
2.4. PROJECT EXPLORER WINDOW .....	31
2.5. PROPERTIES WINDOW .....	32
2.5.1. Properties of multiple object selections.....	32
2.6. TOOLBOX WINDOW .....	33
2.6.1. Toolbox.....	33
2.6.2. Using Engineering Units.....	35
2.7. ALARM VIEWERS .....	35
2.8. PROGRESS BARS .....	63
2.9. BUTTONS.....	77
2.10. DB CONNECTORS.....	93
2.11. DIALOG BOXE.....	125
2.12. DISPLAYS .....	174
2.13. ADVANCED GAUGES AND ARC BARGRAPHS.....	187
2.14. GAUGES .....	200
2.15. FAST METERS .....	212
2.16. HOT REGION .....	216
2.17. DYNAMIC IMAGE OBJECT.....	223
2.18. MULTIMEDIA .....	231
2.19. DRAWS .....	252
2.20. POTENTIOMETERS .....	261
2.21. RECIPE VIEWER.....	272
2.22. SCHEDULER VIEWER .....	280
2.23. SELECTORS.....	287
2.24. VERTICAL AND HORIZONTAL SLIDERS .....	294
2.25. SWITCHES .....	302
2.26. TAB SCREENS .....	310
2.27. TEXTS .....	317
2.28. TRENDS .....	333
2.29. VIEWERS .....	413
2.30. HISTORICAL DATA VIEWERS .....	467
2.31. SYMBOL LIBRARY WINDOW .....	498
2.31.1. Using the Symbol Library.....	498
2.31.2. Creating Composed Symbols.....	500
2.31.3. Power Templates.....	506
2.32. TAG LIST EXPLORER WINDOW .....	510
2.33. OBJECT EXPLORER WINDOW.....	510
2.34. ANIMATION EXPLORER WINDOW .....	511

2.34.1. Dynamic Animation Settings.....	511
2.34.2. Back Color Animation.....	513
2.34.3. Border Color Animation.....	514
2.34.4. Visibility Animation.....	515
2.34.5. Transparency Animation.....	516
2.34.6. Blink Animation.....	517
2.34.7. Blur Animation.....	518
2.34.8. Filling Animation.....	519
2.34.9. Height Animation.....	520
2.34.10. Width Animation.....	521
2.34.11. Scale Animation.....	522
2.34.12. Rotation Animation.....	524
2.34.13. Enable Animation.....	525
2.34.14. Horizontal X Movement Animation.....	526
2.34.15. Vertical Y Movement Animation.....	527
2.34.16. Composed Movement Animation.....	528
2.34.17. Clipping Animation.....	530
2.34.18. Dynamic Borderline Animation.....	531
2.34.19. Dynamic Text Animation.....	532
2.35. COMMAND EXPLORER WINDOW.....	533
2.35.1. Command Settings.....	533
2.35.2. Commands for Screens.....	535
2.35.3. Commands for Tag Variables.....	537
2.35.4. Open Pad Commands.....	541
2.35.5. Recipe Commands.....	543
2.35.6. User Commands.....	544
2.35.7. Change Language Command.....	545
2.35.8. Script Execution Commands.....	545
2.35.9. System Commands.....	546
2.35.10. Commands for Reports.....	547
2.35.11. Alarm Commands.....	549
2.35.12. Logic Commands.....	550
2.35.13. Text to Speech Commands.....	551
2.35.14. Open Map Commands.....	552
2.35.15. Change Converter Commands.....	552
2.35.16. Commands in Methods.....	553
2.35.17. 3D Camera Viewer Commands.....	553
2.36. CLIENT CONNECTIVITY USING AN OPC UA BROWSER.....	554
2.37. OPC UA CLIENT STATUS WINDOW.....	557
2.38. SERVICES CONTROL PANEL.....	558
2.39. DYNAMIC CONFIGURATION - ONLINE CHANGES.....	561

### **3. EDITING PROJECTS ..... 565**

3.1. PROJECT PROPERTIES.....	565
3.2. PROJECT FOLDERS AND FILES.....	568
3.3. TEXT AND MULTILANGUAGES.....	569
3.4. CROSS REFERENCE.....	578
3.5. EVENTS.....	581
3.6. MENU.....	584
3.7. SHORTCUTS.....	590
3.7.1. Shortcut Activation in Runtime.....	592
3.8. CHILD PROJECTS.....	592
3.9. UNIT CONVERTER.....	597
3.10. SPEECH RECOGNITION.....	600
3.11. DATABASE CONFIGURATION TOOL.....	602
3.12. DEPLOYING PROJECTS WITHOUT WEBHMI.....	605

### **4. I/O DATA SERVER..... 609**

4.1. DATA SERVER INTRODUCTION.....	609
4.2. DATA STRUCTURES.....	610



4.3. TAG LIST.....	612
4.3.1. Introduction (Tags).....	612
4.3.2. Tag List.....	613
4.3.3. Tag Properties.....	615
4.3.4. Import/Export.....	621
4.3.5. Server System Tags.....	623
4.4. SETTINGS.....	634
4.4.1. General Server Settings.....	634
4.4.2. Database Connections.....	638
4.4.3. Data Transport Definitions.....	640
4.4.4. Advanced Server Settings.....	643
4.4.5. Server Status and Log.....	648
4.5. GENERAL DRIVER CONCEPTS.....	649
4.6. ALARMS.....	650
4.6.1. Introduction to Alarms.....	650
4.6.2. Alarm Definition Settings.....	651
4.6.3. Alarm Types.....	659
4.6.4. Assigning tags to Alarms.....	667
4.6.5. Viewing Active Alarms.....	672
4.6.6. Historical Log Visualization.....	674
4.7. HISTORIAN.....	675
4.7.1. Introduction Historian.....	675
4.7.2. Historian Settings.....	678
4.7.3. Associating Tags to the Historian.....	682
4.8. DATA LOGGER.....	684
4.8.1. Introduction to using DataLoggers.....	684
4.8.2. Data Logger Settings.....	686
4.8.3. Associating Tags to Data Loggers.....	691
4.9. ENGINEERING UNIT.....	693
4.10. VIEWS.....	697
4.11. REDUNDANCY.....	698
4.11.1. Redundancy (Fault Tolerance).....	698
4.11.2. Redundancy Functionality (Fault Tolerant).....	699
4.11.3. Client Redundancy.....	701
4.11.4. Redundancy Server.....	702
4.12. COMMUNICATION DRIVERS.....	706
4.12.1. General Driver Concepts.....	706
4.12.2. Communication Drivers.....	708
4.13. SERVER (OPC UA BROWSER) DIAGNOSTICS WINDOW.....	711
<b>5. CLIENT CONFIGURATION.....</b>	<b>713</b>
5.1. CLIENT CONNECTIONS TO SERVER.....	713
5.2. CLIENT SETTINGS.....	714
5.3. CLIENT TAG LIST.....	717
5.4. CHANGING APPLICATION NAMES.....	718
<b>6. GRAPHICS.....</b>	<b>721</b>
6.1. SCREENS.....	721
6.1.1. Introduction to Screens.....	721
6.1.2. Inserting new Screens.....	722
6.1.3. Screen Properties.....	725
6.1.4. Screen Navigation.....	733
6.1.5. Embedded Screens.....	735
6.1.6. Screen Layouts.....	735
6.1.7. Pad Screens.....	739
6.1.8. Screen Levels.....	741
6.1.9. Screen Parameterization.....	742
6.1.10. QR Code.....	746
6.2. GRAPHIC EDITING.....	748
6.2.1. Editing Operations.....	748

6.2.2. Graphics Editor.....	752
6.2.3. Color Selection.....	754
6.2.4. Object Contextual Menu.....	760
6.2.5. Importing bitmap images and videos.....	762
6.2.6. Importing vector images.....	764
6.2.7. Static Graphics Effects.....	767
6.2.8. Develop and integrate WPF User Controls in Movicon.NExT.....	768
6.3. ASSIGNING TAGS TO OBJECTS .....	786
6.4. USE OF EXPRESSIONS IN OBJECTS .....	790
6.4.1. Pointing to Variable Bits and Array Variable Elements.....	790
6.4.2. Other Expressions.....	792
6.5. ALIASES IN OBJECTS .....	824
6.6. CLIENT SYSTEM TAGS .....	830
6.7. STYLES.....	832
6.7.1. Using Styles.....	832
6.7.2. Managing Styles in Movicon Next.....	833
6.8. 3D GRAPHICS .....	842
6.8.1. 3D Graphics.....	842
6.8.2. 3D Animation Graphics.....	846
6.8.3. 3D Symbol Properties.....	850
6.8.4. Inner Screens for 3D Animation Graphics.....	852
6.8.5. Using the Camera in 3D Graphics.....	854
6.9. MULTITOUCH.....	855
6.9.1. Using multitouch.....	855
6.9.2. Multi Touch Gestuality.....	857
6.10. GEOSCADADA .....	858
6.10.1. Geo Page.....	858
6.10.2. Screens in Geographic Maps.....	861
6.10.3. Alternative Providers and Local Maps.....	864
6.10.4. Maps in WebHMI.....	866
<b>7. RECIPE MANAGER.....</b>	<b>869</b>
7.1. INTRODUCTION TO USING RECIPES .....	869
7.2. RECIPE MANAGEMENT .....	871
7.3. RECIPE IMPORT & EXPORT .....	873
7.4. RECIPE PROPERTIES .....	874
7.5. USING RECIPES AT RUNTIME .....	879
<b>8. SCHEDULER MANAGER .....</b>	<b>883</b>
8.1. INTRODUCTION TO USING SCHEDULERS .....	883
8.2. GENERAL SCHEDULER SERVER SETTINGS.....	885
8.3. ADVANCED SCHEDULER SETTINGS.....	887
8.4. TYPES OF SCHEDULERS .....	889
8.5. SCHEDULER PROPERTIES .....	893
8.6. THE SCHEDULER VIEWER OBJECT.....	897
<b>9. SCRIPTS VB.NET .....</b>	<b>899</b>
9.1. INTRODUCTION .....	899
9.2. SCRIPT EDITOR.....	900
9.3. BASIC SCRIPT DEBUG .....	903
9.4. BASIC SCRIPT VARIABLES .....	905
9.5. BASIC SCRIPT TYPES .....	908
9.5.1. Script Typology.....	908
9.5.2. Script Resource.....	909
9.5.3. Basic Script Tags.....	915
9.5.4. Code in Basic Scripts and in Symbols.....	917
9.5.5. Basic Script Libraries.....	919
<b>10. LOGIC .....</b>	<b>921</b>

10.1. INTRODUCTION TO USING LOGIC .....	921
10.2. LOGIC FUNCTION BLOCKS AND GATES .....	921
10.3. ASSIGNING TAGS.....	924
10.4. LOGIC RESOURCE PROPERTIES .....	925
10.5. LOGIC EXECUTION .....	927
<b>11. REPORTS .....</b>	<b>929</b>
11.1. INTRODUCTION TO USING THE REPORT DESIGNER.....	929
11.2. REPORT DESIGNER.....	930
11.3. REPORT PROPERTIES .....	931
11.4. REPORT EXECUTION.....	933
11.5. REPORT VIA WEB .....	935
11.6. PASSING PARAMETERS AND DB CONNECTIONS .....	937
<b>12. ALARM DISPATCHER.....</b>	<b>943</b>
12.1. INTRODUCTION TO USING THE ALARM DISPATCHER .....	943
12.2. ALARM DISPATCHER SETTINGS .....	944
12.3. ADVANCED ALARM DISPATCHER SETTINGS.....	946
12.4. ALARM DISPATCHER PLUGIN LIST.....	949
12.5. ALARM DISPATCHER NOTIFICATIONS .....	949
12.6. PLUGIN .....	954
12.6.1. Telegram Sender.....	954
12.6.2. Smtip Mail Sender.....	961
12.6.3. Sending Vocal Messages.....	963
12.6.4. GsmSMS Sender.....	966
<b>13. USERS &amp; SECURITY .....</b>	<b>969</b>
13.1. GENERAL SECURITY CONCEPTS .....	969
13.2. SECURITY MANAGEMENT.....	969
13.3. PROJECT PROTECTION AND ENCRYPTION .....	973
13.4. USERS AND PASSWORDS MANAGEMENT .....	974
13.4.1. General User Management Settings.....	974
13.4.2. User Management.....	980
13.4.3. User Properties.....	982
13.4.4. User Group Properties.....	984
13.4.5. User Commands.....	985
13.4.6. Domain User Authentication.....	986
13.5. CFR21 PART11 .....	987
13.5.1. General Concepts.....	987
13.5.2. Security.....	989
13.5.3. Validation and Documentation.....	991
13.5.4. CFR21 part 11 configuration.....	993
13.5.5. Data Backup Validation.....	999
13.5.6. User Sharing.....	1001
13.5.7. Trace/Audit for Project User Management.....	1002
<b>14. WEBCLIENT.....</b>	<b>1003</b>
14.1. INTRODUCTION TO USING THE WEBCLIENT .....	1003
14.2. WEBCLIENT (WPF/HTML5) .....	1003
14.2.1. Web Server Architecture.....	1003
14.2.2. IIS Web Server Configuration.....	1005
14.2.3. Client Limitations.....	1007
14.2.4. Project Deployment.....	1008
14.2.5. Web Page menu.....	1014
14.2.6. Managing Client Users.....	1017
14.2.7. Client Errors.....	1022
14.3. WEBCLIENT (APP) .....	1022
14.3.1. Deploying Projects.....	1022
14.3.2. APP Installation and Configuration.....	1026

14.3.3. App Limitations.....	1030
14.4. WEBHMI (SVG/HTML5).....	1031
14.4.1. Introduction to using the WebHMI.....	1031
14.4.2. WebHMI Client.....	1033
14.4.3. Integrated Web Server Settings.....	1036
14.4.4. WebHMI Functions and Limitations.....	1038
<b>15. OPC UA .....</b>	<b>1041</b>
15.1. OPC UA INFORMATION MODEL IN PLATFORM.NEXT .....	1041
15.2. OPC UA SECURITY.....	1044
15.3. OPC UA CERTIFICATES.....	1047
15.4. OPC UA SERVER.....	1050
15.4.1. OPC UA Server.....	1050
15.5. OPC UA CLIENT.....	1054
15.5.1. OPC UA Client.....	1054
15.5.2. Client-side connection via OPC UA browser.....	1056
15.5.3. Connecting to Server with an OPC UA Client communication driver.....	1058
<b>16. MOVICON WEBHMI .....</b>	<b>1063</b>
16.1. INTRODUCTION TO USING THE WEBHMI.....	1063
16.2. WEBHMI RUNTIME WINDOWS.....	1065
16.3. WEBHMI RUNTIME LINUX (x64) .....	1068
16.4. WEBHMI RUNTIME LINUX (ARM) .....	1071
16.5. DEPLOYING PROJECTS IN HMI .....	1074
16.6. INTEGRATED WEB SERVER SETTINGS.....	1080
16.7. SECURITY CERTIFICATE MANAGEMENT .....	1081
16.8. WEBHMI FUNCTIONS AND LIMITATIONS .....	1083
16.9. HOW TO ACTIVATE A WEBHMI RUNTIME LICENSE.....	1085
16.10. COGNITIVE AUGMENTED REALITY .....	1089
16.10.1. Introduction.....	1089
16.10.2. Cognitive Augmented Reality System Architecture.....	1091
16.10.3. Project and Movicon.NEXT WebHMI Configuration.....	1093
16.10.4. TensorFlow Installation and Configuration.....	1095
16.10.5. Python Script Installation and Configuration .....	1101
<b>17. RUNTIME MANAGER .....</b>	<b>1105</b>
17.1. AUTOMATIC PROJECT STARTUP FROM COMMAND LINE .....	1105
17.2. STARTING PROJECT UP IN EDITOR .....	1106
17.3. CLICKONCE CLIENT STARTUP .....	1108
17.4. STARTING UP AS SERVICE .....	1109
17.5. RUNTIME DIAGNOSTICS .....	1112
17.5.1. Connection Status.....	1112
17.5.2. Client Status (SHIFT + F1).....	1112
17.5.3. System Log (SHIFT + F2).....	1113
17.5.4. Cross Reference (SHIFT + F3).....	1114
17.5.5. Watch (SHIFT + F5).....	1115
17.6. PC TIME CHANGES .....	1116
<b>18. LICENSE MANAGER.....</b>	<b>1117</b>
18.1. INTRODUCTION TO USING LICENSES .....	1117
18.2. I/O TAG COUNT .....	1118
18.3. INSTALLED LICENSE VERIFICATION .....	1120
18.4. LICENSE MANAGER .....	1124
18.5. NET LICENSES.....	1127
18.6. WEBCLIENT LICENSES.....	1131
18.7. LICENSE ACTIVATION .....	1131
<b>19. MOVICON NEXT BUILDER .....</b>	<b>1135</b>
19.1. INTRODUCTION .....	1135

19.2. VISUAL STUDIO PROJECT SETTINGS.....	1136
19.3. PROJECT INSTANCE MANAGER.....	1137
19.4. TEXTRESOURCE .....	1137
19.4.1. AddLanguage, TextResource Method .....	1137
19.4.2. AddString, TextResource Method.....	1138
19.4.3. DeleteLanguage, TextResource Method.....	1139
19.4.4. DeleteString, TextResource Method.....	1139
19.4.5. GetString, TextResources Method.....	1140
19.4.6. Save, TextResource Method.....	1140
19.4.7. SetString, TextResource Method.....	1141
19.4.8. TextResource, TextResource Method.....	1141
19.5. REPORTCONTAINER.....	1142
19.5.1. GetParameterValue, ReportContainer Method.....	1142
19.5.2. GetReport, ReportContainer Method.....	1142
19.5.3. ReportContainer, ReportContainer Method.....	1143
19.5.4. Save, ReportContainer Method.....	1143
19.5.5. SetParameterValue, ReportContainer Method.....	1143
19.6. SCREENCONTAINER.....	1144
19.6.1. AddFolder, ScreenContainer Method.....	1144
19.6.2. AddGeoData, ScreenContainer Method.....	1145
19.6.3. AddScreen, ScreenContainer Method.....	1145
19.6.4. AddSymbol, ScreenContainer Method.....	1146
19.6.5. AddToolboxItem, ScreenContainer Method.....	1147
19.6.6. DeleteElement, ScreenContainer Method.....	1148
19.6.7. DeleteScreen, ScreenContainer Method.....	1148
19.6.8. GetItem, ScreenContainer Method.....	1149
19.6.9. GetScreen, ScreenContainer Method.....	1153
19.6.10. GetScreenUri, ScreenContainer Method.....	1153
19.6.11. NewScreenTypesFolder, ScreenContainer Method.....	1154
19.6.12. Save, ScreenContainer Method.....	1154
19.6.13. Screen Container Examples.....	1155
19.6.14. ScreenContainer, ScreenContainer Method.....	1156
19.6.15. SearchXamlItem, ScreenContainer Method.....	1156
19.6.16. SetBackground, ScreenContainer Method.....	1156
19.6.17. SetScreenEntity, ScreenContainer Method.....	1157
19.6.18. ToolBoxFolder, ScreenContainer Method.....	1158
19.7. IODATASERVER.....	1158
19.7.1. AddAlarmArea, IODataServer Method.....	1158
19.7.2. AddAlarmDefinition, IODataServer Method.....	1159
19.7.3. AddAlarmSource, IODataServer Method.....	1160
19.7.4. AddBaseAddress, IODataServer Method.....	1160
19.7.5. AddDatalogger, IODataServer Method.....	1161
19.7.6. AddDataLoggerColumn, IODataServer Method.....	1162
19.7.7. AddEngineeringUnit, IODataServer Method.....	1163
19.7.8. AddFolder, IODataServer Method.....	1163
19.7.9. AddHistorian, IODataServer Method.....	1164
19.7.10. AddNewDriver, IODataServer Method.....	1165
19.7.11. AddPrototype, IODataServer Method.....	1165
19.7.12. AddPrototypeMember, IODataServer Method.....	1166
19.7.13. AddStation, IODataServer Method.....	1167
19.7.14. AddTag, IODataServer Method.....	1168
19.7.15. AssignAlarmToTag, IODataServer Method.....	1169
19.7.16. DeleteAlarmArea, IODataServer Method.....	1169
19.7.17. DeleteAlarmDefinition, IODataServer Method.....	1170
19.7.18. DeleteAlarmSource, IODataServer Method.....	1171
19.7.19. DeleteDatalogger, IODataServer Method.....	1171
19.7.20. DeleteDataLoggerColumn, IODataServer Method.....	1172
19.7.21. DeleteEngineeringUnit, IODataServer Method.....	1172
19.7.22. DeleteFolder, IODataServer Method.....	1173
19.7.23. DeleteHistorian, IODataServer Method.....	1174

19.7.24. DeletePrototype, IODataServer Method.....	1174
19.7.25. DeleteTag, IODataServer Method.....	1175
19.7.26. GetAlarmArea, IODataServer Method.....	1175
19.7.27. GetAlarmDefinition, IODataServer Method.....	1176
19.7.28. GetAlarmSource, IODataServer Method.....	1176
19.7.29. GetBaseAddress, IODataServer Method.....	1177
19.7.30. GetChannel, IODataServer Method.....	1177
19.7.31. GetDataLogger, IODataServer Method.....	1178
19.7.32. GetDataLoggerColumn, IODataServer Method.....	1178
19.7.33. GetDataLoggerList, IODataServer Method.....	1179
19.7.34. GetDriverConfiguration, IODataServer Method.htm.....	1179
19.7.35. GetEngineeringUnit, IODataServer Metod.....	1180
19.7.36. GetFolder, IODataServer Metod.....	1180
19.7.37. GetHistorian, IODataServer Method.....	1181
19.7.38. GetHistorianList, IODataServer Method.....	1181
19.7.39. GetPrototype, IODataServer Method.....	1182
19.7.40. GetServerConfiguration, IODataServer Method.....	1182
19.7.41. GetServerEntityReference, IODataServer Method.....	1183
19.7.42. GetStation, IODataServer Method.htm.....	1183
19.7.43. GetTag, IODataServer Method.....	1184
19.7.44. GetTagEntityReference, IODataServer Method.....	1184
19.7.45. GetTagOPCUAEntityReference, IODataServer Method.....	1185
19.7.46. GetUFATag, IODataServer Method.....	1186
19.7.47. GetVarTagEntityReference, IODataServer Method.....	1187
19.7.48. IODataServer, IODataServer Method.....	1188
19.7.49. RemoveAlarmFromTag, IODataServer Method.....	1189
19.7.50. RemoveChannel, IODataServer Method.htm.....	1189
19.7.51. RemoveDriver, IODataServer Method.htm.....	1190
19.7.52. RemoveStation, IODataServer Method.htm.....	1190
19.7.53. Save, IODataServer Method.....	1191
19.8. PARAMETERS.....	1191
19.8.1. AddParameterFile, Parameter Method.....	1191
19.8.2. DeleteParameterFile, Parameter Method.....	1192
19.8.3. GetParameterFile, Parameter Method.....	1192
19.8.4. GetParameterUri, Parameter Method.....	1193
19.8.5. Parameters, Parameter Method.....	1193
19.8.6. Save, Parameter Method.....	1194

## 20. TECHNICAL SPECIFICATIONS..... 1195

20.1. REQUIREMENTS.....	1195
20.2. GENERAL SPECIFICATIONS.....	1198
20.2.1. Users and Groups.....	1198
20.2.2. Text and String Resource.....	1199
20.2.3. Windows Services.....	1200
20.2.4. Logic.....	1200
20.2.5. Child Projects.....	1201
20.2.6. Alarm Dispatcher.....	1201
20.2.7. Schedulers.....	1202
20.2.8. Networking.....	1202
20.2.9. Scripts.....	1203
20.3. WEB CLIENT SPECIFICATIONS.....	1204
20.3.1. Web Client.....	1204
20.3.2. WebHMI Functions and Limitations.....	1205
20.4. CLIENT SPECIFICATIONS.....	1207
20.4.1. Screens.....	1207
20.4.2. Networking.....	1210
20.4.3. OPC UA Client.....	1211
20.4.4. Menu.....	1211
20.4.5. Shortcuts.....	1212
20.4.6. Multi Touch.....	1212

20.5. SERVER SPECIFICATIONS .....	1213
20.5.1. Tags.....	1213
20.5.2. Alarm and Event Log.....	1215
20.5.3. Engineering Units.....	1217
4.1.1. Historian .....	1217
4.1.2. Recipes .....	1219
4.1.3. Drivers .....	1219
4.1.4. OPC UA Server.....	1221
4.1.5. Events.....	1221
4.1.6. Redundancy.....	1222





## 1. 1. Introduction

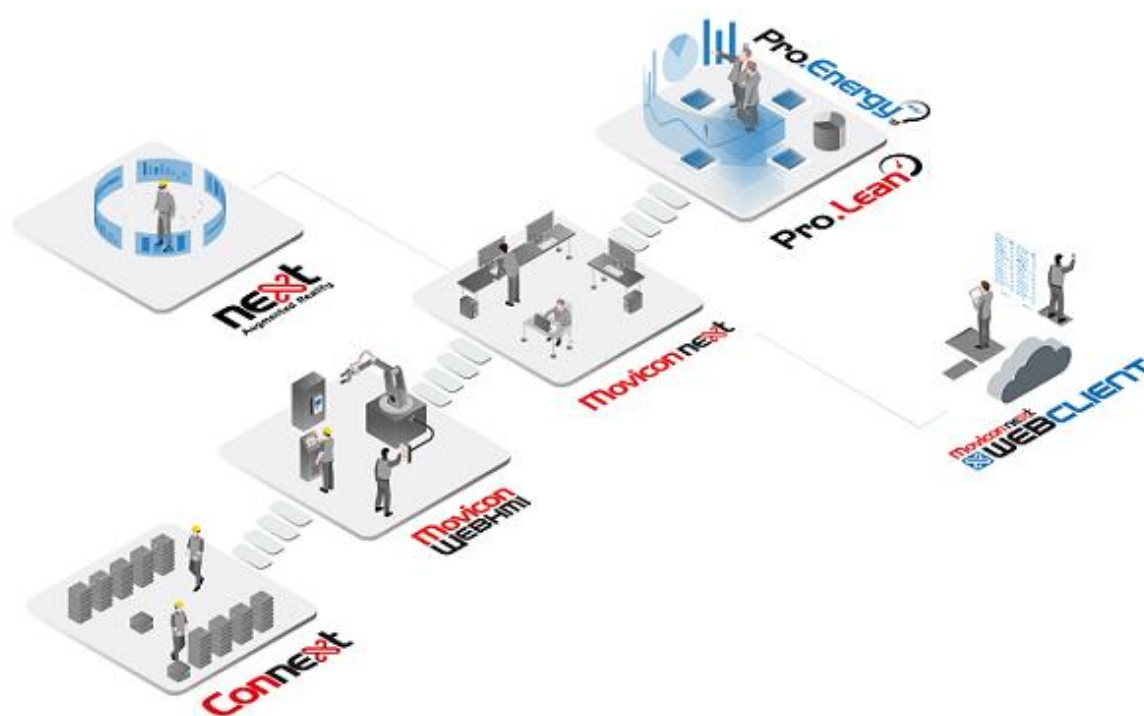
## 1.1. Movicon.NExT & WebHMI

Welcome to the Movicon.NExT™ software platform.

Movicon.NExT™ is a software architecture designed to build platforms of modern automation systems. It surpasses the restricted use of conventional Scada/HMI technologies by proposing new generation software solutions natively based on the most modern technological solutions. It has been designed primarily to allow users to create the foundations for long term investments without compromise by offering the openness and integration never thought possible before in the world of automation.

Movicon.NExT™ is an open and scalable platform based on .NET and modern technology based on modular concepts with plugin technology to enhance industrial software architectures with more openness and scalability.

This platform is the ideal solution for those companies who are looking to build the foundations for future-proof automation systems using cutting-edge software technology based on .NET, C#, WPF/XAML, OPC UA, SQL Server and HTML5.



## Modularity and Scalability

The Movicon.NExT platform is modular and scalable to ensure the required openness to integrate your every automation need. It offers:

1. An all-in-one intuitive editor environment that makes it easier to create even the most complex projects that integrates the system's functional models with each other, or those created by third parties and easy to plugin. The same Editor can be used to create projects that can be launched from the Movicon.NExT runtime (PC Windows) and WebHMI runtime (Linux HMI architecture).

2. Effective, safe, and performing project runtime execution for SCADA/HMI architectures for PCs and Embedded PCs, based on Windows 64 bit (Movicon.NExT Runtime)
3. Entry-level HMI project runtime execution for applications on HMI panels with Web-based architecture, for hardware solutions based on Linux systems, such as Raspberry and ARM, or Windows, using the .NET Core technology.

### Power and Openness

The Movicon.NExT software platform has been especially designed to quickly create SCADA/HMI projects by configuring resources and properties in an object-oriented working environment.

The editor allows you to create the most powerful and effective functionalities relating to field device communication, visualization, alarm management, data recording, powerful analysis functions, security management, control logic, enterprise-wide information distribution, locally or geographically distributed. All this within a hand's reach in one platform that offers the great advantage of total integration and plug-in modularity.

### Documentation Scope

The analytic approach of this guide aims to give you a quick 'n' easy to follow demonstration on how to use the platform's features.

This guide is accompanied by a "Tutorial" with step-by-step learning to help you get to grips with the basics.

All the information provided in this guide presupposes that:

- The user has sufficient know-how of the Windows 7, Windows 8 or Windows 10 techniques
- The user has sufficient know-how of operation systems, Tag and PLC concepts
- The user has the basic know-how of file and database management

## 1.2. Project Architecture

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

The architecture uses a Client - Server model based on a .NET framework created specifically for industrial systems.

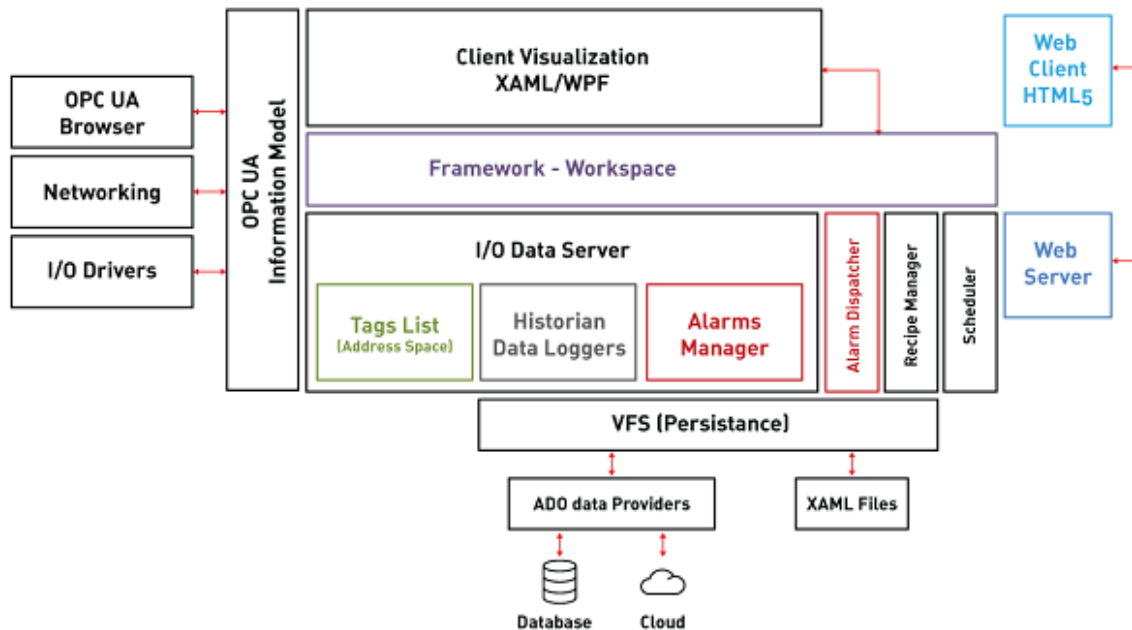
The projects are principally built one part Server (for managing communications, data, alarms and historical logging) and one part Client (for managing the Movicon.NExT graphical displays). The platform's information model is based on the OPC UA technology and totally transparent to users. Therefore a Server project provides its Variables for full disposal by means of publishing them as OPC UA tags in the Address Space enabling the Client project to access the list of these Variables according to its assigned access rights. The Movicon.NExT modularity enables Movicon Clients to connect to third party OPC UA Servers without using Server side, or vice-versa enabling third party Client connection to the Movicon.NExT Server application.

The Platform NExT 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.

Below is a simplified schemata of the Movicon project architecture:



*Simplified Movicon project architecture.*

Movicon.NExT uses the most innovative technologies such as "OPC UA", "WPF", "WCF".

## 1.3. Important Information

### 1.3.1. Preface

All the information provided in the Movicon documentation is based on the assumption that the following points have been reviewed:

- All the illustrations in this guide refer to the software version indicated in the documentation with Windows™32/64 bit platform.
- Windows refers to the registered Microsoft inc. trademark for the 32/64 bit and successive versions.
- Ms Access, SQL Server and Azure refer to registered Microsoft inc. products.
- Movicon.NExT™ refers to software products developed and protected by international copyright laws.
- Any other product or trademark cited whether registered or protected by proprietary rights.
- Windows has been installed correctly in the system. For further information relating to the installation procedures, please consult the appropriate sections in the Windows "Introduction Guide".



#### **Important**

All the information contained in this guide is subject to modification without prior warning due to system updates.

### 1.3.2. Warranty and Responsibility



#### **Warranty limitations**

This software product is not guaranteed for use in conditions where system reliability is in close connection to the safety and protection of persons and property, or where any defects might cause direct or indirect harm and damage to persons or property, such as medical surgery and operation theatres, nuclear power plants, military bases or other. The reliability of each software application, including the present one, may be subjected to alteration deriving and due to several factors, which include power supply, computer hardware problems, operating system malfunction, electrical disturbances, program errors, application and installation errors, overheating, malfunctioning of connected electrical devices, incorrect use and mishandling by user or project designer. All these factors are defined as "system failures" here within this document. Therefore each application where a "system failure" may cause serious damage to objects and people (injury risk or life threatening) should implement security systems that are not based solely on electronic systems, considering the risks of electronic "system failure" as a whole.



#### **Responsibility of the user**

It is the responsibility of the user or project designer to verify that the "system" is suitable to the purposes they wish to accomplish. Progea shall not in anyway whatsoever be liable for damages to property or persons as specified in the GUARANTEE LIMITATIONS accepted by user when installing and using software product.



#### **Disclaimer of All Warranties**

- PROGEA DISCLAIMS ANY AND ALL WARRANTIES WITH REGARD TO Movicon.NExT AND THE RELATED DOCUMENTATION, WHETHER EXPRESSED OR IMPLIED. PROGEA SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF FITNESS FOR A GENERAL OR PARTICULAR PURPOSE, MERCHANTABILITY, SATISFACTORY QUALITY AND THOSE ARISING OUT OF COURSE OF DEALINGS OR USAGE IN TRADE.
- Movicon.NExT AND THE RELATED DOCUMENTATION ARE PROVIDED "AS IS," AND YOUR COMPANY UNDERSTANDS THAT IT ASSUMES ALL RISKS OF THEIR USE, QUALITY, AND PERFORMANCE.



#### **Disclaimer of Liability**

YOUR COMPANY AGREES AND ACKNOWLEDGES THAT PROGEA SHALL HAVE NO LIABILITY WHATSOEVER TO YOUR COMPANY FOR ANY PROBLEMS IN OR CAUSED BY MOVICON OR THE RELATED DOCUMENTATION, WHETHER DIRECT, INDIRECT, INCIDENTAL, SPECIAL, OR CONSEQUENTIAL (INCLUDING LOSS OF PROFITS).



#### **Using Script code**

Warning: The use of script code in Movicon projects is subject to a LIMITED WARRANTY. The User has complete freedom to manage script when using events and methods of the product and of third party components. However, Progea may not be able to fully guarantee the correct behaviour of the product in the event of using script code in properties or specific events or methods. Therefore Users are strongly advised to always refer to the documentation whenever possible and take preventive measures by checking that the code they intend to use functions correctly. However, Progea does provide a LIMITED WARRANTY for those project functionalities that use VB.NET.

### 1.3.3. Copyright

Connext© and Movicon NExT© are software products and a copyright of Progea. All rights are reserved. The Movicon™ brand name is a registered trademark of Progea.

Progea srl  
Via D'Annunzio 295,  
41123 Modena, Italy

Under the copyright laws, this publication shall not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of Progea.

### 1.3.4. EndUser License Agreement (EULA)

#### Important!

*Read carefully before installing this software*

This is a legally binding agreement between the buyer (the "User") of the software product (the "Software"), e.g. **Movicon.NExT™**, and Progea Srl, a company which is part of Emerson Automation Solutions ("Emerson"). Before continuing with the installation of the software, the User must read, acknowledge and accept the following terms and conditions stipulated in this User License Agreement (the "Agreement") for the Software. This Agreement can also be found in printed form inside the package containing the Software's media and accessories. By opening the Software package, indicating assent electronically, or downloading, installing, copying, or using the Software, the User accepts the terms and conditions of this Agreement. If the User does not accept the terms and conditions of this Agreement, the User must return the purchased product with all its components, including any associated media (printed materials and online or electronic documentation, binders and any other component), within 30 days of purchase, to the original reseller for a full refund, or write to: Progea Srl, via G. D'Annunzio 295 - 41123 Modena - Italy, to receive return instructions.

## Software user license

### 1. License terms

Subject to the User's compliance with this Agreement and payment of applicable fees, Emerson grants the User a single and non-exclusive license (the "License") to use a single copy of the Software product, including any updates if free, and additional modules or components purchased in conjunction therewith, solely for the User's own internal business purposes, and solely in accordance with the terms and conditions of this Agreement. Failure to pay the applicable license fees immediately terminates the License grant to the User. The User may copy the Software to the memory of another computer only as necessary to use the Software in accordance with this Agreement. The User's license to use the Software may be perpetual or subject to a subscription period as specified in the documentation governing the User's purchase of the license. User agrees not to exceed any limits applicable to use of the Software (e.g., as may be identified in the documentation for the Software) nor use any features of the Software for which the User has not purchased a license. Emerson may use technological and/or other measures to prevent unlicensed access to and/or use of the Software and/or its features, and the User acknowledges

and agrees to the use of such measures, and that the User shall not circumvent nor attempt to circumvent such measures. EXCEPT AND ONLY TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE USER MAY NOT REVERSE ENGINEER, DECOMPILE, DISASSEMBLE, OR ATTEMPT TO DISCOVER THE SOURCE CODE OF THE SOFTWARE OR ANY COMPONENTS THEREOF, INCLUDING GRAPHIC SYMBOLS.

## 2. Choice of Software

The Users declares to have chosen the Software after assessing, before purchase, its features and specifications, taking into account the usage requirements and characteristics of any computer on which the Software is to be used. The User is solely and fully responsible for this assessment and therefore for the choice made. The purpose of the Software referred to in this Agreement, according to the Software's features and technical specifications, is to carry out the operations indicated in the user's manual only.

## 3. Installation and use

The Software can be used in one of five possible configurations: (1) single user License (2) multi-user License (3) upgrade License (4) evaluation License (5) OEM License. The User shall install and use the Software in only one of the configurations for which the License has been granted. The Software License is granted individually to one single User. The License may be fully transferred to another User, but it must not be shared and used by more than one User at the same time.

- Single user License. The User may install and use one single licensed copy of the Software on one single computer. The User may access the Software from a hard disk or over a network, but the User must only operate on one single computer, according to the terms and conditions of this Agreement.
- Multiple user License. The User must acquire a multiple user License to use the Software on two or more workstations on a network. The number of users acquired with the multi-License corresponds to the maximum number of computers on which the Software can be used simultaneously. The Software can be installed on any number of computers on the same network, but the number of simultaneous sessions of the Software must not exceed the number of users specified in the multi-License.
- Upgrade License. The User may use one copy of the Software purchased as upgrade, provided that the User has already purchased and installed a copy of the previous version of the original Software. The upgrade License is expressly restricted to the replacement of the previous version of the Software with the equivalent upgraded Software product. Additional or multiple user licenses cannot be acquired as upgrade licenses.
- Evaluation License. The User may use a copy of the Software on one single computer, for a maximum period of ninety (90) days. The User has limited rights to use the Software for evaluation and testing purposes only, with possible features limitations. The User does not acquire any warranty or other License rights over the use of this Software in evaluation mode.
- OEM License. If the User has acquired the Software License as pre-installed on a computer or otherwise in combination with a third-party hardware product, the User has purchased an OEM License. The OEM License is equivalent to a single license for a single computer. If not specifically authorized by the hardware supplier, the User does not automatically acquire a License to use the Software. The Software can be accessed locally on the hardware or over a network, but it must only be used on a single computer.

## 4. Copyright and copies

The Software (including any associated component) is the exclusive property of Emerson and is protected by Italian copyright laws, international trademark protection rules and international copyrights laws and treaties. The User only receives a License to use the Software according to the terms and conditions in this Agreement, therefore the User does not acquire any ownership rights, neither over the Software, nor its source codes. The User is authorized to make one copy of the Software solely for:

- creating one backup copy for archival purposes only,
- transferring the Software to one single hard disk with the purpose of creating a backup copy.

The User must not make and/or use copies of the Software for any purpose except those explicitly permitted by the applicable laws. Furthermore, the User must not copy or reproduce any electronic documentation, associated media or printed material accompanying the Software. Emerson hereby reserves all legal rights, including any right not explicitly indicated in this Agreement.

## **5. Other restrictions**

The Software and all its associated media, subject of this Agreement, remain the exclusive property of Emerson.

The User is explicitly forbidden from:

- using the Software to provide hosted services, training services, consulting services, service bureau, time sharing, or rental use to third parties without written permission from Emerson; and
- duplicating the Software and/or its documentation other than for the permitted or authorized purposes.

The User may permanently transfer the Software, including any previous versions of it, all accompanying printed materials, the License to use the Software, this Agreement, and any copy of all the above to an operator or owner (or a subsequent operator or owner) of the facility or equipment at which the Software is installed, provided the recipient fully agrees to the terms of this Agreement without any exceptions.

In all cases, the User must transfer all rights and any warranties to the third-party recipient. The User only has the right to use the Software subject of this Agreement according to Art.5 (Art.64-Bis and ter) in the Italian State Law no.518, effective since 29 December 1992, and no.248/00.

## **6. Limited warranty and remedy**

Emerson warrants the media on which the Software is furnished to be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of delivery to the User. However, Emerson does not warrant that the functions contained in the Software will meet the User's requirements or that the operation of the Software will be uninterrupted or error free. In addition, Emerson does not warrant that the Software or any equipment, system, or network on which the Software is used will be free of vulnerability to intrusion or attack.

## **7. No other warranties**

The warranty and remedy stated in the foregoing article are Emerson's sole and exclusive warranty obligation and the User's sole and exclusive remedy. THE SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. EMERSON MAKES NO REPRESENTATIONS OR WARRANTIES ON BEHALF OF ITS LICENSORS.

## **8. Limitations of liability**

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT AND FOR NO REASON SHALL EMERSON NOR ITS AFFILIATES, LICENSORS, AND RESELLERS BE LIABLE FOR ANY DAMAGE, EXPENSE OR COST DERIVING FROM THE REPAIR OR REPLACEMENT OF PRODUCTS, OR FOR ANY CONSEQUENTIAL, SPECIAL, INCIDENTAL, PUNITIVE OR INDIRECT DAMAGE ARISING FROM THE USE OR INABILITY TO USE THE SOFTWARE, ASSOCIATED MEDIA OR ACCOMPANYING DOCUMENTATION, EVEN IF EMERSON HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT, ANY RESIDUAL MANDATORY OR APPLICABLE LIABILITY TO EMERSON, ITS AFFILIATES, LICENSORS OR RESELLERS SHALL EXCEED THE PRICE PAID BY THE USER TO PURCHASE THE LICENSE.

## **9. Resolution, termination of agreement and license revocation**

Emerson retains the right to revoke the License for the use of the Software and terminate this Agreement with immediate effect by sending written notice to the User by registered post signed on arrival, pursuant article 1456 of the Italian Civil Code, in the event the User:

- fails to comply with the provisions at articles 3), 4), 5), 10) and 11) of this Agreement.
- is declared Insolvent.



In case of termination of this Agreement due to any of the above events, Emerson reserves the right to claim compensation for direct and/or indirect, incidental or consequential damages, including compensation for additional losses, caused in relation to the breach of this Agreement, and based on the extent to which the provisions of this Agreement were unfulfilled. Upon termination of this Agreement or License revocation, including decision by the User to terminate or relinquish the License for any reason, the User shall return the full Software package, including any dongle, media and accessory documentation to Emerson, including in cases where the Software has been modified and/or integrated into other products. The User shall promptly destroy any remaining copy of The Software, including modifications, integrated portions, updates and upgrades in any form, and provide immediate evidence by written notification to Emerson.

#### **10. Information material protection - industrial and intellectual property.**

All the information material that comes into the possession of the User pursuant to the Agreement is protected by copyright rules and laws. All registered and unregistered trademarks, as well as any and all other distinctive signs or names appearing in the Software and/or media and/or the printed or digital accompanying documentation, remain the property of Emerson. The stipulation of this Agreement does not transfer or give to the User any rights of ownership to the Software or to any related patents, copyrights, trademarks or other intellectual property in any way whatsoever.

#### **11. Data Privacy and use**

When the User downloads, installs or uses the Software, Emerson may use automatic means to collect information about the User's device and about the User's use of the Software. The User also may be required to provide certain information about the User as a condition to downloading, installing or using the Software or certain of its features or functionality, and the Software may provide the User with opportunities to share information about the User with others. Emerson may collect information, including but not limited to information relating to the manner or frequency of use of the Software. This includes data that Emerson collect as part of our services. Some of the data Emerson collects is relatively static, such as data about the product (for example, serial numbers and device IDs and the Emerson model and serial number), but other data is more dynamic, such as information about current and past product usage levels, information about abnormal product functioning, information about the status and location of products used with the Software and/or the User's device. Emerson may collect this data from the User and the Software and transmit it through the User's network and/or Internet services. Monitoring may continue until the Software is deleted or the services are cancelled or terminated. Note that if the User does not allow Emerson to collect this information, Emerson may not be able to provide the full capabilities of products and services. All information Emerson collects through or in connection with this Software and the use of the Software in connection with Emerson products is subject to the Emerson privacy notice found at <https://www.emerson.com/en-us/privacy-notice> ("the Privacy Notice"). Emerson will collect, use and disclose the User's personal information and the personal information of the User's additional users as set out in that Privacy Notice, and the User and the additional users or third parties the User has granted access or control consent to Emerson doing so. The Privacy Notice applies subject to requirements of local law such as General Data Protection Regulation (EU 2016/679 and Italian Law Decree 101/2018. In the event of a conflict between the Privacy Notice and applicable data protection laws, the stricter obligation applies. Furthermore, the data that is generated and collected by the Software may be stored in a hosted environment at a location identified to the User in the documentation for the Software. The User is solely responsible for compliance with all applicable local, state and federal data sovereignty laws, regulations and restrictions as it relates to the movement of data generated and collected by the Software.

#### **12. Security assessments and digital signatures**

In the event the User wishes to assess the vulnerability of the Software (including conducting a penetration test or any other form of vulnerability assessment or test), the User agrees to obtain Emerson's prior written consent. Emerson will not unreasonably withhold or unduly delay such consent, but the User agrees that Emerson may require that: (i) Emerson participate in any such tests/assessments,



(ii) receive the results of such tests/assessments, and (iii) the User not publish, distribute, or otherwise make available the results (in whole or part) to any third party without the express written consent of Emerson. Emerson may further require that any such tests/assessments be conducted in accordance with and subject to a separate written agreement between the User and Emerson.

Emerson may attach or otherwise associate digital signatures to files of the Software to aid detection of subsequent modifications to such files. However, in attaching or associating such digital signatures, Emerson does not (i) guarantee that the Software or the files can be trusted, nor (ii) assume any liability or claim authorship with regard to third party files.

### **13. Third-party software**

To the extent that a third-party owns any portion of the Software licensed under this Agreement, such third-party owner shall be a beneficiary of this Agreement and shall have the right to enforce its rights under this Agreement independently of Emerson. Any third-party Software, not licensed under this Agreement, shall be subject to third-party owner's applicable license agreement and registration requirement, and the User authorizes Emerson to provide the User's contact information, including company name, mailing address, telephone number, ship-to recipient address, and primary contact's name, mailing address, telephone number, and email address, to such third-party owner for the purposes of product registration, support, and compliance with applicable import and export control laws, regulations, orders, and requirements.

### **14. Export compliance**

The User agrees that all applicable import, export control and sanctions laws, regulations, orders and requirements, as they may be amended from time to time, including without limitation those of the United States, the European Union and the jurisdictions in which Emerson and the User are established or from which items may be supplied, and the requirements of any licenses, authorizations, general licenses or license exceptions relating thereto will apply to its receipt and use of the Software. In no event shall the User use, transfer, release, export or re-export the Software in violation of such applicable laws, regulations, orders or requirements or the requirements of any licenses, authorizations or license exceptions relating thereto. If there is a change in any applicable laws, regulations, orders or requirements that would prohibit Emerson from providing the Software, or would in the reasonable judgement of Emerson otherwise expose Emerson to a risk of liability under such laws, regulations, orders or requirements if it provided the Software, Emerson shall be relieved of any obligation to provide the Software and/or any related services without penalty.

### **15. Nuclear applications**

The Software is not provided or intended for use in any nuclear or nuclear related applications. The User accepts the Software with the foregoing understanding and agree to indemnify and hold harmless Emerson from any claims, losses, suits, judgments and damages, including incidental and consequential damages, arising from such use, whether the cause of action be based in tort, contract or otherwise, including allegations that Emerson's liability is based on negligence or strict liability.

### **16. Miscellaneous**

This Agreement is the only binding agreement between the User and Emerson and supersedes any prior verbal or written agreement relating to the Software License or products purchased by the User. Both parties renounce the application of the United Nations Convention on International Sales of Goods. This Agreement is governed by the current Italian laws and exclude the possibility of reference or conflicts with different laws. All disputes arising out of this Agreement shall be litigated or otherwise resolved exclusively in Italy at the Court of Justice located in Milano. If any provision of this Agreement is ruled invalid, such invalidity shall not affect the validity of the remaining portions of this Agreement. The use of the Software, that is the opening of the package containing the physical support with the Software and the installation of the Software on a computer, determines the execution of the License Agreement and the complete acceptance of the provisions in the Agreement by the User, without any reservation and/or

exception. The User declares to qualify as a legal business and not as a consumer and declares to have chosen the Software with the purpose of using it according to the provisions of this Agreement for professional and/or business reasons and not for personal use.











Except as expressly authorized in this Agreement, the User may not sublicense, assign, or transfer the License or the Software without the prior written consent of Emerson. Any attempt otherwise to sublicense, assign or transfer any of the rights, duties, or obligations hereunder without such consent is void.


























The User agrees that Emerson may audit the User's facility and records to confirm compliance with the terms of this agreement. Emerson agrees to provide a reasonable written notice prior to conducting such an audit and agrees to conduct the audit during regular business hours without unreasonably interfering with the User's normal business operation.























If the User does not accept the terms and conditions of this Agreement, the User must return the purchased product with all its components, including any associated media (printed materials and online or electronic documentation, binders and any other component), within 30 days of purchase, to the original reseller for a full refund, or write to: Progea Srl, via G. D'Annunzio 295 - 41123 Modena - Italy, to receive return instructions

### 1.3.5. Open Source Components

Movicon.NExT uses third party components whose licenses and/or copyright notices are provided below. Note: Website addresses in this section are from the respective companies at time of publication and subject to change at owners' discretion.






















Library Name	Version	Author	License Type	License
3DTools	1.0.1	Not Available	MS-LPL	
Ookii.Dialogs.Wpf	1.0.0	Sven Groot	BSD 3-Clause	
HelixToolkit.Wpf	2012.1.1	helix-toolkit.org	MIT	
MjpegProcessor	1.1.0	Brian Peek	MIT	
GadgetLibrary	1.0.0	Ron Gramann	CPOL	
Autoupdater.NET	1.3.1	rbsoft.org	MIT	
WpfKb	0.1.0	imasm	MS-PL	
jQuery SignalR		.NET Foundation	Apache-2.0	
jquery-validation-unobtrusive	3.2.11	.NET Foundation	Apache-2.0	
Amib.Threading	2.2.3	Ami Bar	Ms-PL	
svg-pan-zoom	3.6.0	Andrea Leofreddi	BSD 2-Clause	
Aga.Controls	1.0.0	Andrey Gliznetsov	Apache 2.0	
Bootstrap	4.3.0	Bootstrap	MIT	
jQuery Mousewheel	3.1.12	Brandon Aaron	MIT	
Simple CommandLine Parser	1.0.0.0	Chris Stoy	CPOL 1.02	





















jQuery hashchange event	1.3	Cowboy Ben Alman	MIT & GPL	
Weak Events		Daniel Grunwald	MIT	
String.format for JavaScript		Daniel Mester Pirttijärvi	Zlib	
jQuery Kinetic	1.5	Dave Taylor	MIT	
LinqStatistics	1.0.0	Don Kackman	Apache-2.0	
ObservableDictionary		Dr. WPF	MIT	
jquery-layout	1.2.0	Fabrizio Balliano & Kevin Dalman	MIT & GPL	
Modernizr JavaScript library		Faruk Ates	BSD & MIT	
angular	1.7.7	Google	MIT	
i18next	3.0.0	i18next	MIT	
i18next-browser-languagedetector	2.2.2	i18next	MIT	
i18next-xhr-backend	2.0.0	i18next	MIT	
jquery-i18next		i18next	MIT	
howler	2.1.2	James Simpson of GoldFire Studios	MIT	
jQuery Underscore	1.3.3	Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors		
jquery-validation	1.19.1	Jörn Zaefferer	MIT	
Thriple		Josh Smith	Ms-PL	
RequireJS	2.3.6	jQuery Foundation	MIT	
jQuery JavaScript		JS Foundation	MIT	
Globalize	1.4.0	JS Foundation	MIT	
jquery-ui		JS Foundation	MIT	
Moment		JS Foundation	MIT	
ShaderEffects	1.0.0	Kubajzz	CPOL 1.02	
nouislider	13.1.4	Léon Gersen	MIT	
msgpack5	4.2.1	Matteo Collina	MIT	

jQuery floatThead	2.1.2	Misha Koryak	MIT	
LogViewer		Olivier Dahan & Ken C. Len	Ms-RL	
OPCUA Libraries		OPC Foundation	RCL	
html-imports		Polymer Project	BSD 3-Clause	
Simple CLDR traverser (cdnjs)	19.0.3	Rafael Xavier	MIT	
jQuery UI Virtual Keyboard	1.30.1	Rob Garrison & Jeremy Satterfield	MIT	
FFmpeg.Autogen	3.3.2	Ruslan Balanukhin	LGPL 3	
jQuery knockout-sortable		Ryan Niemeyer	MIT	
jQuery Knockout JavaScript	2.2.1	Steven Sanderson	MIT	
JSZip		Stuart Knightley	MIT & GPL	
detect-zoom	1.0.4	Tom Bigelajzen	WTFPL & MIT	
Unosquare Ffmpeg MediaElement	3.3.2	Unosquare	Ms-PL	
svg.filter.js	3.0.4	Wout Fierens	MIT	
svg.js	3.0.12	Wout Fierens	MIT	
jQuery SignalR		.NET Foundation	Apache-2.0	
DotRas	1.2.0	Jeff Winn	LGPL 3	
Path-data-polyfill		Jarek Foksa	MIT	
jinit	2.11.58	Sebastien ros	BSD 2-Clause	
JS Foundation	4.0.1	JS Foundation	BSD 2-Clause	
Steve Towner	1.0.0	Steve Towner	MIT	
PDF Reader in JavaScript (pdf.js)	2.9.359	Various contributors	Apache 2.0	
esprima	4.0.1	JS Foundation	BSD 2-Clause	
ScintillaNET.WPF	1.0.0	Steve Towner	MIT	
jsqr	1.3.1	Cosmo Wolfe	Apache 2.0	

## Packages





















As regards to the packages, the libraries are used as they are provided without being modified and without publishing any source files.

Library Name	Version	Author	License Type	License
CsvHelper	12.0.0	Josh Close	Apache2.0	
Owin	1.0	OWIN startup components contributors	Apache2.0	
DotNetZip	1.9.3	Henrik/Dino Chiesa	MS-PL	
jQuery	3.1.1	jQuery Foundation, Inc.	MIT	
log4net	2.0.8	The Apache Software Foundation	Apache2.0	
MarkerMetro.Unity.Ionic.Zlib	2.0.0.14	MarkerMetro	MIT	
MySql.Data	8.0.13	Oracle	-	
Nager.Date		Tino Hager	MIT	
NetTelegramBotApi	3.7.50926	Dmitry Popov	MIT	
Telegram.Bot	13.4.0	RoundRobin, Poulad	MIT	
Newtonsoft.Json	12.0.3	James Newton-King	MIT	
PropertyChanged.Fody	2.6.0	Simon Cropp	MIT	
RestSharp		John Sheehan, Andrew Young, Alexey Zimarev and RestSharp community	Apache2.0	
System.Data.SQLite	10.0.111.0	SQLite Development Team	-	
System.Data.SQLite.EF6	10.0.111.0	SQLite Development Team	-	
TLSharp	0.1.0.339	Ilya P	MIT	
FodyHelpers		FodyHelpers	MIT	
knockoutjs	3.5.1	The Knockout.js team	MIT	
NetMQ	4.0.0.207	NetMQ	GLGPL	
Portable.BouncyCastle	1.8.9	Oren Novotny	-	
System.Data.SQLite.Core	1.0.111.0	SQLite Development Team	-	

System.Data.SQLite.Linq	1.0.111.0	SQLite Development Team	-	
MouseKeyHook		George Mamaladze	MIT	
M2Mqtt	4.3.0.0	Paolo Patierno	Eclipse Public License 1.0	
AmazedSaint.ElasticObject	1.0.0	amazedsaint	Ms-PL	
System.Reactive.Core	4.1.3	.NET Foundation and Contributors	Apache-2.0	
System.Reactive.Interfaces	4.1.3	.NET Foundation and Contributors	Apache-2.0	
System.Reactive.Linq	4.1.3	.NET Foundation and Contributors	Apache-2.0	
System.Reactive.PlatformServices	4.1.3	.NET Foundation and Contributors	Apache-2.0	
Microsoft.Toolkit.Wpf.UI.Controls.WebView	6.0.0	.NET Foundation and Contributors	MIT	
BigMath	0.5.0	Alexander Logger	MIT	
TaskScheduler	2.8.18	David Hall	MIT	
ExcelDataReader	3.6.0	ExcelDataReader developers	MIT	
M2MqttDotnetCore	1.0.8	Hamidreza Mohaqeq	EPL	
Mindscape.Raygun4Net.AspNetCore	6.2.0	Raygun	MIT	
Mindscape.Raygun4Net.NetCore	6.0.0	Raygun	MIT	
System.Buffers	4.5.1	Microsoft	MIT	
MimeKit	2.10.1	Jeffrey Stedfast	MIT	
MailKit	2.10.1	Jeffrey Stedfast	MIT	
estebano.ScintillaNET.WPF	2.6.0	Steve Towner	MIT	
jacobslusser.ScintillaNET	3.6.3	jacob slusser	MIT	

## APP Packages

The following table shows the packages related to the Movicon.NEXt APP

Library Name	Version	Author	License Type	License
PCLCrypto	2.0.147	Andrew Arnott	Ms-PL	
PInvoke.Bcrypt	0.5.152	Andrew Arnott	MIT	
PInvoke.Kernel32	0.5.152	Andrew Arnott	MIT	
PInvoke.Ncrypt	0.5.152	Andrew Arnott	MIT	
PInvoke.Windows.Core	0.5.152	Andrew Arnott	MIT	
Validation	2.4.18	Andrew Arnott	MIT	
SQLitePCLRaw.bundle_green	1.1.11	Eric Sink	Apache-2.0	
SQLitePCLRaw.core	1.1.11	Eric Sink	Apache-2.0	
SQLitePCLRaw.provider.sqlite3.ios_unified	1.1.11	Eric Sink	Apache-2.0	
sqlite-net-pcl	1.4.118	Frank A. Krueger	MIT	
FastAndroidCamera	2.0.0	James Athey	Apache-2.0	
Newtonsoft.Json	10.0.3	James Newton-King	MIT	
modernhttpclient	2.4.2	Paul Betts	MIT	
ZXing.Net.Mobile	2.3.2	Redth	Apache-2.0	
ZXing.Net.Mobile.Forms	2.3.2	Redth	Apache-2.0	
SQLitePCLRaw.lib.e_sqlite3.android	1.1.9	Eric Sink, D. Richard Hipp	Apache-2.0	
SQLite.UWP.Native	3.12.3	SQLite Development Team	Public Domain	
Xamarin.Forms	2.5.1.527436	Xamarin Inc.	MIT	
Xamarin.Forms.Pages	2.5.1.527436	Xamarin Inc.	MIT	
SecureStorage	2.5.0	Sameer Khandekar	MIT	

### 1.3.6. Software Product Lifecycle Policy

A timeline is defined to determine the availability of support and maintenance for a specific version of a Movicon software product. This policy is used by customers and partners to plan utilization and upgrade strategies for a software product.

## Definition of Terms:

- **Support** is access to product experts that can help end-users troubleshoot and solve problems regarding the application of an Emerson software product.
- **Maintenance** is software releases that include bug fixes and enhancements.
- **Version** is a specific release of a product that uses a numeric designation to identify it as a major release. Currently different schemes are used to designate a major release. The aim is to use the same designation for all MAS software.
- **Versioning** is the sequence of numbers separated by dots to identify any different releases of the software.

Example: Movicon <X>.<Y>.<ZZZ> where:

X = Major Version (usually released every 3-5 years, and generally requires a license upgrade.

Y = Minor Version (usually released every year and is generally free of charge. It includes bug fixes and product improvements.

ZZZ = Build: (usually released 2-3 times a year free of charge and includes only maintenance bug fixes)

## Classifying Product Versions to Indicate Available Level of Support and Maintenance

### Classifying Product Versions to Indicate Available Level of Support and Maintenance

With the passage of time, newer versions of a software product become available. The new versions are compatible with new operating systems and hardware and include features to meet the emerging needs of end-users. These new versions eventually replace older versions.

This process of releasing new product versions over time creates a need to classify the different versions of a product to indicate the level of support and maintenance associated with them. To meet this need, Emerson has created the active, mature, and retired classifications.

- **Active** is the actual version in production. Support is active and maintenance is active through the updated number of Builds.
- **Mature** is one of the previous minor versions which is no longer in production but still fully supported. Support is active and Maintenance is granted under certain conditions.
- **Retired** is one of the previous major versions. Support is no longer available or not guaranteed, and updates are not provided.

## Lifecycle Classifications for Product Versions

Version Classification	Definition	Usage Guidelines	Support	Maintenance	Training
<b>Active</b>	Latest released version of a software product	New system installations, major expansion	Primary Support	Maintenance releases for bug fixes and product enhancements are provided. Validation provided for new MS operating systems.*	Available
<b>Mature</b>	The version replaced by the latest version, or a previous version less	Expansion where consistency is more important than new features	Primary Support	No bug fixing or enhancements are included in these releases (due to free update to Active versions). Typically, Maintenance Hot Fixes can be only provided	Available



	than 4 years old or needed for compatibility	and future support		under business justification No validation for new MS operating systems.	
<b>Retired</b>	A previously released version	Systems where bug fixes, operating system support, and security updates are not a concern.	No Support. Special Support Contract can be agreed without warranties of Hot Fixes	No maintenance releases.	Not offered

\*this is for new MS Major Operating Systems, for example Win10, Win11, and include Windows updates, for example Win10 20H2

## Software Versions Available for Ordering and Downloading

Only the Active version of a product can be ordered. This is because licenses are backwards compatible, so a license will support the active version at the time of purchase, and all previously released versions. A license upgrade is necessary to move to a higher or more current version.  
All active and mature versions are available for download from the Customer Center.

## Movicon Versioning Scheme

### Product Versions

Movicon products use a sequence of numbers separated by dots to distinguish different releases of the software products. Specifically, when a software product has a new release, it will be given a unique sequence of numbers where the first number in the sequence identifies the major version. The second number identifies minor releases, and the third number identifies a simple maintenance update, for that version.

Example: Movicon.NExT 3.2.184

A license upgrade is usually needed to move from one "Major release" to another. License upgrading is to be established according to the corresponding sales policies.

### Maintenance Releases for a Version

Each version has maintenance releases, that is identified by a "build" number which is the last number in the versioning number sequence. A maintenance release with bug fixes is made available when necessary, according to the maintenance policy in question. These releases are always free of charge and considered as maintenance releases.

### Release Notes

Each version is described in a release note, which is also available on the [www.progea.com](http://www.progea.com) web site. Release notes state reports all the changes made for each version. Any software changes are identified with a case number for product traceability and incompatibility purposes. Customers are strongly advised to read these release notes before installing new versions.

## Current Movicon Software Status

Version	State	Release Date	Mature Date	Retired Date	Latest Maintenance Release
2.x	Retired	Nov. 24, 2014	June 15, 2016	Nov. 24, 2018	2.2.33 - Released on April 5, 2016
3.x	Retired	June 15, 2016	June 01, 2020	June 01, 20121	3.4.271 - Released on May 11, 2020
4.0	Mature	June 01, 2020	June 01, 2021	TBD	4.0.300 - Released on Jan. 20, 2021
4.1	Active	June 01, 2020	TBD	TBD	4.1.329 - Released on Oct. 6, 2021
4.2	To be released				

### 1.3.7. Waste Disposal

SGLOCK USB dongles must be disposed of in a dedicated Waste Disposal bank according to national waste legislation and regulations. The symbol shown on the SGLOCK dongle indicates that this item must be disposed of in a "separate waste disposal bank".



Therefore users must use their nearest waste disposal facilities provided by the local authorities or hand it in to their local dealers when purchasing another of its kind. Waste disposal collection and the following treatment, recovery and disposal operations enables the manufacturing of devices with recycled materials and environmental protection against any negative effects and health hazards caused by improper disposal. Any improper disposal of this device by the user will be subject to legal sanctions under Article 254 and pursuant Legislative Decree no. 152 of April 3 2006.

Pursuant to art. 13 of Legislative Decree of 25th July 2005, n. 151 "Implementation of Directives 2002/95/EC, 2002/96/EC and 2003/108/EC concerning the reduction of using hazardous substances in electrical and electronic equipment and waste disposal".

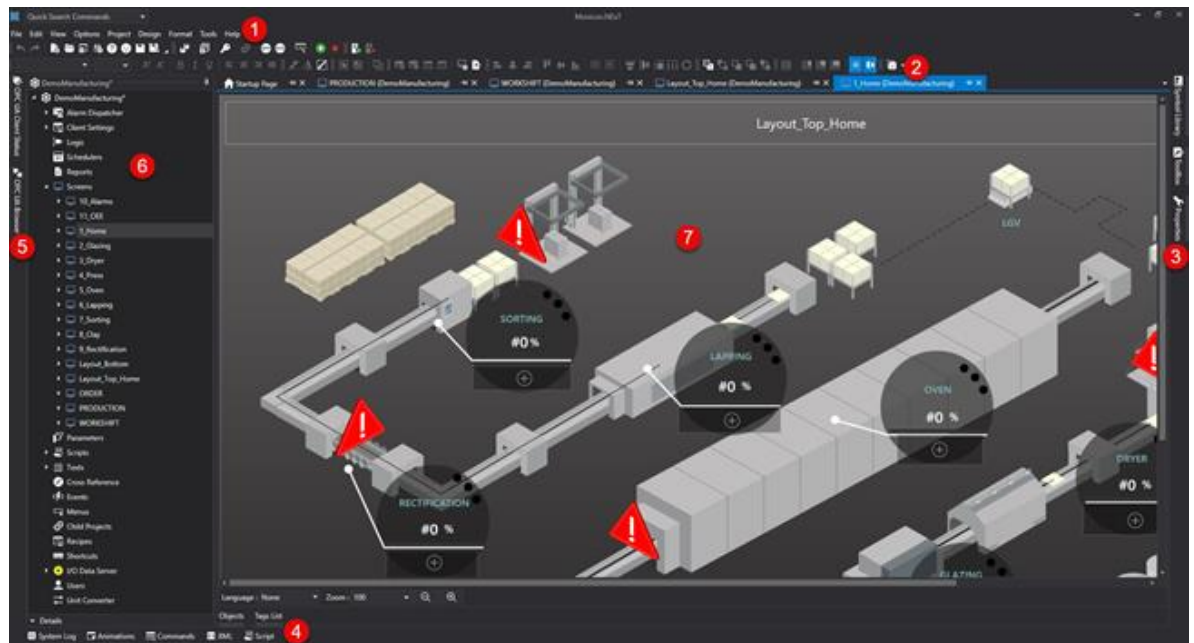




## 2. The development environment

### 2.1. Workspace

The Movicon.NExT workspace has been designed according to the most modern ergonomics concepts and offers a user interface composed of panels that can be organized as needed.



*Screenshot of the Movicon.NExT workspace.*

The main features shown in the workspace are:

- The Menu
- The customizable Toolbars
- The "Proprieties", "Toolbox" and "Symbol Gallery" panels
- The "System Log", "XML Explorer", "Animation Explorer", "Command Explorer" and "Script Explorer" panels
- The "OPC UA Browser" and "OPC UA Client Status" panels
- The Project Explorer
- The Editing area

The workspace panels (3, 4 and 5) can be managed according to these three modes:

- Floating: this is a default setting and represents the window as a Tab on the top workspace border that when clicked will activate the window.
- Docked: clicking the window's "pin" icon, will dock the window rendering it visible in the workspace.
- Popup: When dragging the window into the workspace, it will no longer remain docked to the border but managed as a popup window. To dock the window again within the workspace, simply drag it towards the border.

### Easy-mode

The Easy-Mode function simplifies the use of the workspace for the user by hiding windows that are used for more advanced purposes. Basically, this command is available from the toolbar or the "View" menu and deactivates the following windows:

- OPC UA Client Status
- OPC UA Browser
- System Log
- XML Explorer

In addition to these, the 'Child Projects' resource, from the project's tree structure, will also be banned from use.

## Workspace Configuration Management

When changes have been made to the workspace, such as moving the properties window or making them bigger, they will stay that way when restarting Movicon.NExT.

In addition to this type of automatic saving of the user interface, there also exists the possibility to manually save specific workspace configurations.

This possibility allows you to manually save a configuration of windows in the editor to load it at a later time. This will allow you, for example, to load a particular type of configuration of windows when working with a notebook rather than loading a different one when the notebook connect to an external monitor.

This management can be enabled by setting the 'File mode' option present in the File -> Workspace Layout menu.

### Save layout configuration

This command is used to save the current editor's configuration so that it can be called at a later time.

To save the layout process as follows:

- Adjust the editor windows as desired by moving them or changing their sizes.
- Make sure the 'Workspace Layout' -> 'File mode' option is enabled in the 'File' menu.
- From the same 'File -> Workspace Layout' menu select the 'Save Layout' command.
- Enter the name you wish to assign to the configuration in the 'Name' field of the popup window that appears.
- Press 'OK'.

As a result the new layout will be saved and inserted within the grid underneath.

The summary grid contained within the window also allows you to filter or search for the layout displays. You can use the search field at the top right of the window to search for the desired layout configuration by entering its name.

As regards to filtering the layout display, use the "pen" icon at the bottom right of the grid.

### Loading a layout configuration

In case where a layout configuration has already been created, it can be called by the user to quickly reset the windows within the editor.

When loading a certain type of layout configuration and modifying its workspace aspects manually, its original aspects can be restored by reloading its configuration again.

Proceed in this way to load a layout:

- Make sure the 'Workspace Layout' -> 'File mode' option is enabled in the 'File' menu.
- From the same 'File -> Workspace Layout' menu select the 'Load Layout' command.
- Select the desired configuration from the grid.

- Press 'OK'.

The summary grid contained in the window also allows you to filter or reload layout configuration displays.

You can use the search field at the top right of the window to search for the desired layout configuration by entering its name.

As regards to filtering the layout display, use the "pen" icon at the bottom right of the grid.

### Import and export one or more existing configurations

The import or export of configurations saved in a machine towards another can be done by manually copying/pasting the folder that contains the configuration files from the Windows resource explorer. The path where the layout configuration are to be saved is:

*C:\ProgramData\Progea\Movicon.NExT.Layouts*

Inside this folder there are other subfolders whose names are those of the configurations saved with the editor. To import a layout configuration from a different computer proceed as follows:

- Using the resource explorer, open the C:\ProgramData\Progea\Movicon.NExT.Layouts folder.
- Copy all the contents from the folder or select the subfolders of the configurations desired.
- Paste the subfolders selected in the C:\ProgramData\Progea\Movicon.NExT.Layouts folder present in the target computer.

### Remove existing layout configurations

In order to remove an existing layout configuration, you can use either the window for saving or loading configurations.

Basically, once the window for loading configurations is open, you can select one of the items presented with the grid and press the CANCEL button.

To search for configuration, you can use the search field at the top right of the window by entering the name of configuration you are looking for.

As regards to filtering the layout display, use the "pen" icon at the bottom right of the grid.

### Restoring original layout configuration

All the changes made to the windows that have been used to compose the layout are saved and stored for the next Movicon.NExT startup.

However, it may be handy to be able to restore the sizes/visualizations of the screens to their original state.

There are two ways in which the workspace can be restored:

#### Startup with CTRL Key

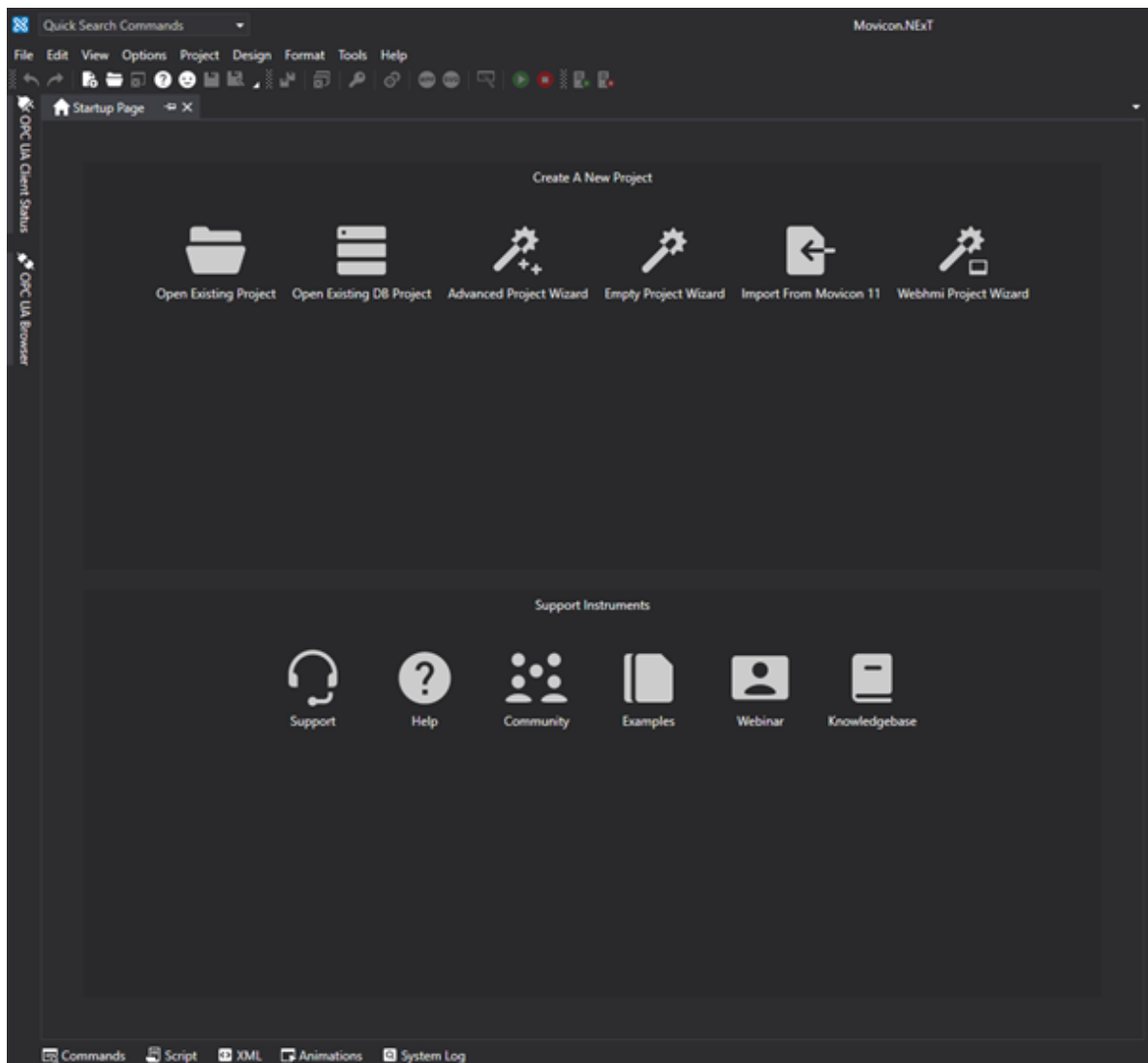
- Close all the Movicon.NExT instances that have been started up.
- Restart Movicon.NExT while keeping the keyboard's CTRL key pressed down until the editor has been completely loaded.

#### Deleting Isolated Storage

- Close all the Movicon.NExT instances that have been started up.
- Open a Resource Explorer window at the following location %LocalAppData%\IsolatedStorage and delete the folders.

## 2.2. Startup Page

Launching the development environment will display the "startup Page", which is a workspace through which the designer can open the project desired or use to access information about the platform.



The Startup Page allows:

- Access to open and create project commands using the wizard.
- Access to key tools that are very handy for the designer to use such as Guidelines, Forums, Examples, Webinar, Knowledge Base.
- The list of the last opened projects through which the most recent one can be opened directly with a simple click.

### Opening a Project

The "Open" command opens an existing project. The projects can be based on XML file and on Database. For this reason, the system will request the designer to specify the existing project's data source by selecting one of these options:

- **File Source:** selection of a normal XML project file by browsing files in the PC folders.
- **Data Source:** Selection of a project residing on database by accessing the database according to the desired database connection string.



The project data will then be saved encoded within a table structure. When choosing to save the project on file in xml format, Movicon will create a project folder within which the project information will be saved in a file structure and in subfolders. The structure of the subfolders is similar to the one used for the project's resources displayed in the Movicon.NExT "Project Window". Once confirming the command, the project will open in the workspace.

## Project Creation Wizard

The "Create New" command is used to create a new project by selecting one of the three project model types:

- Local Project: Server and Client running on the same machine.
- Distributed Project: Server and Client running on different machines.
- Project with Redundant Servers: More than one server is present with the Redundancy option.



The "Distributed Project" model also supports Clients running on the same machine where Server is running.

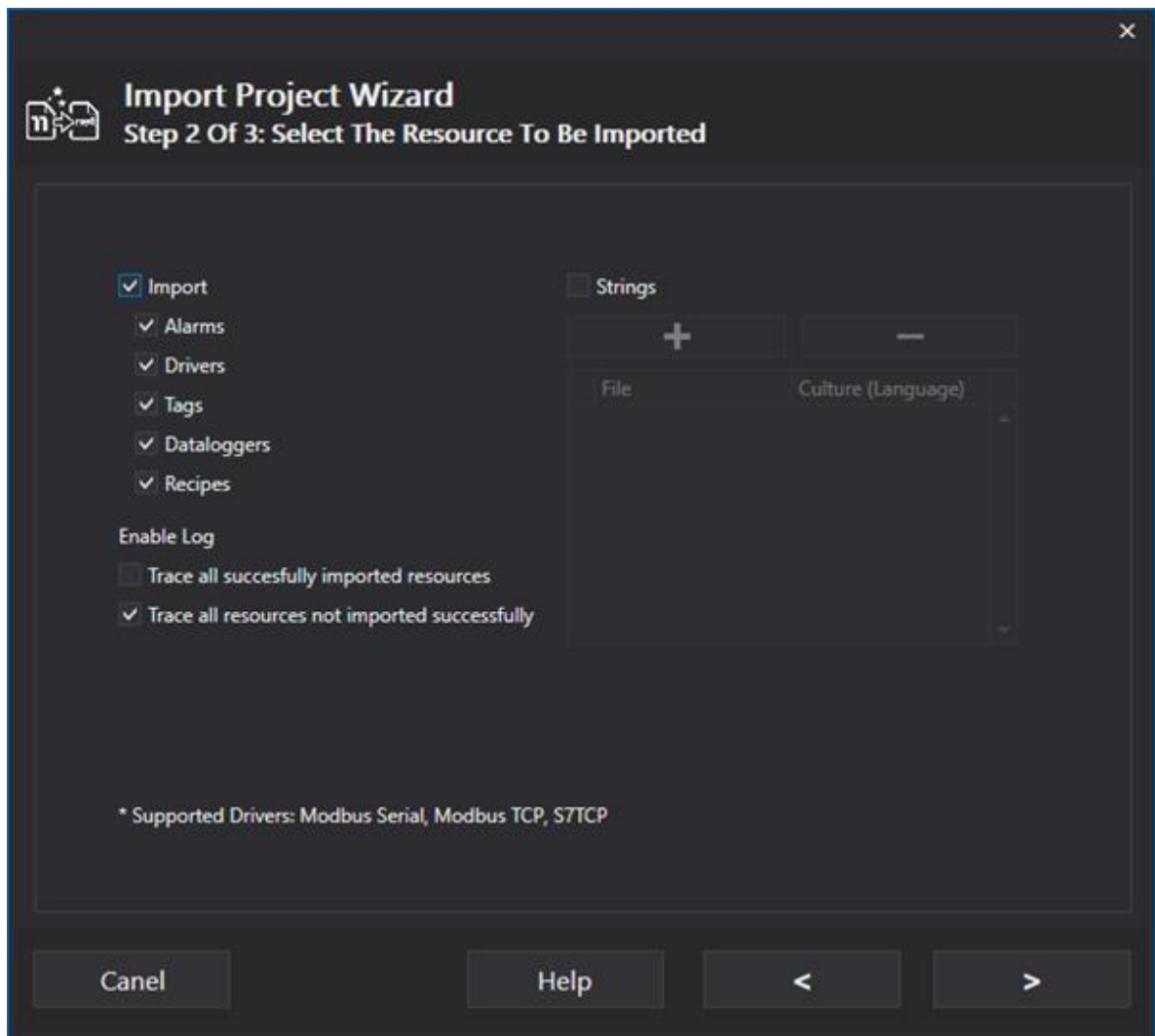
After selecting the model type, the guided procedure will help the developer to define the name of the project and the target path (on file or database).

The number and type of Screen to add to the project along with the Alarm Definition and historian definitions can be specified in the advanced wizard.

## Movicon 11 Project Import Wizard

The "Import Project" command permits resources to be imported from Movicon11 projects by using a wizard that guides the user with a simple step-by-step procedure.

Once the Movicon 11 project has been selected, it will be possible to then select which resources to import into the new Movicon.NExT project: Alarms, Drivers, Tags, Dataloggers, Recipes, Strings.



## Alarms

The Exclusive alarm types are converted to, in groups of four thresholds, ExclusiveLevel alarm types. The thresholds of each remaining alarms are converted to TripAlarm types.

The thresholds with Between condition types are not supported.

The thresholds with RateChangeIncrease or RateChangeDecrease condition types are converted to ExclusiveRateOfChange alarm types.

The ReadAccessLevel and WriteAccessLevel parameters are ignored in the import phase.

## Tags and Structure Tags

Simple tags and structure tags are both supported (in this case, the relative Data Structure is created).

When the Tag import terminates, they are then associated to the alarms based on the alarms imported.

The Physical I/O Addresses are imported if the corresponding driver is supported.

## Drivers



Only the Master TCP Modbus, Serial Modbus and Siemens S7 TCP drivers are currently supported.

The static tasks are associated to the relative tags as Physical I/O Addresses. The importer calculates the start address for each tag based on its size.

Static tasks defined in individual structure members are not supported.

The Movicon 11 RegisterSize parameter is not yet supported in tasks in the MoviconNExT ModbusTCP driver, therefore 2 byte records are read by default and a station is created for each Movicon 11 ServerAddress (station) and for each UnitID defined in the tasks.

The SwapByte value for the 2,3 and 5 FunctionCode is inverted 2, 3 e 5 because it is applied in the driver code by default in Movicon.NExT.

### Datalogger

The Dataloggers of the source project are imported with the exception of the dataloggers in InMemoryDB or RecordOnChange types.

When the source datalogger is defined with a specific DSN, it will be necessary to configure the connection string manually in Movicon.NExT

### Recipes

When importing recipes, the columns defined in the recipe structure are first imported then those defined using the recipe as template.

For each column in the source recipe, a group is created along with the corresponding Data Value with the start address defined in any existing reference variable.

The Description, Measure Unit properties and the Data Value default value properties based on the reference tag of the source column are defined in the import phase.

### Strings

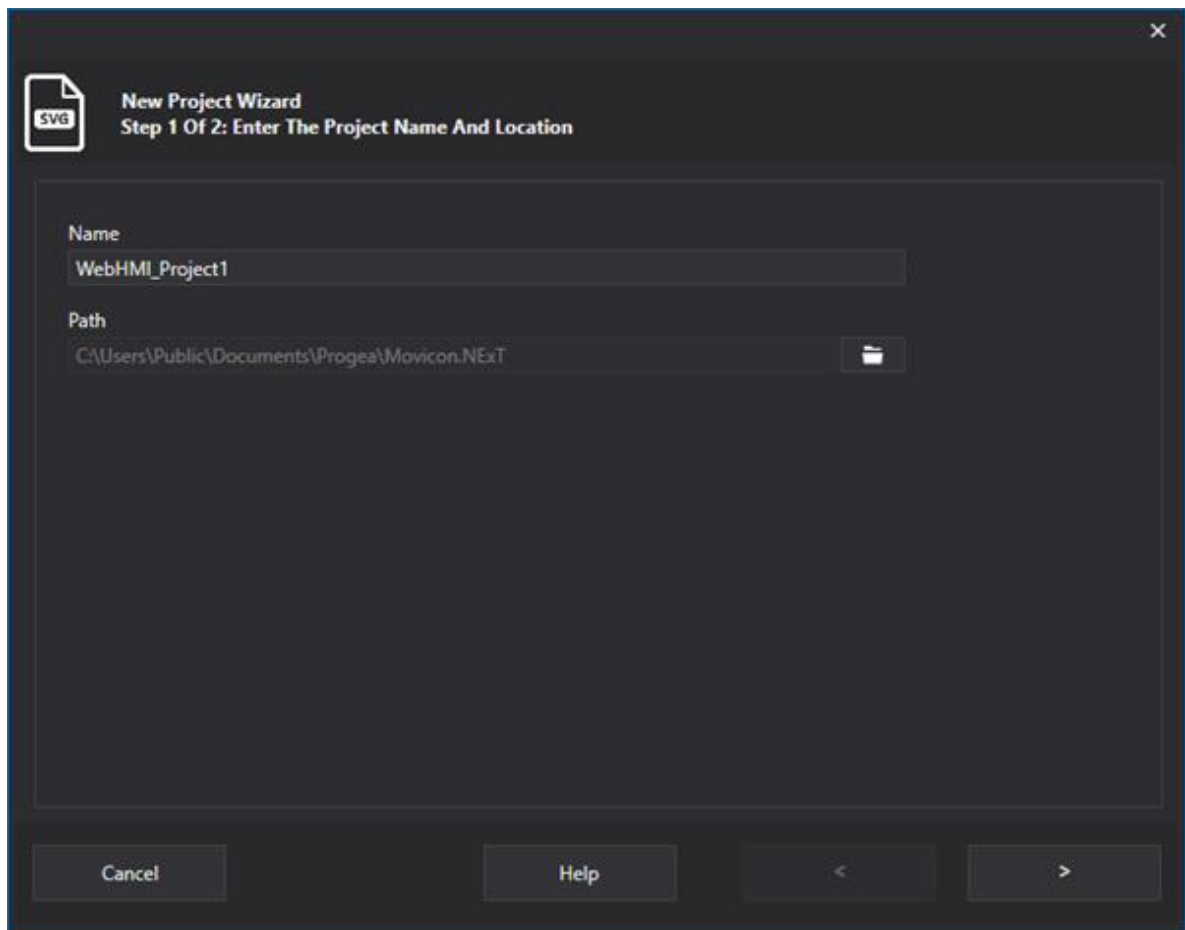
The import of text used in the project is done by means of selecting the

<ProjectName>string.<ColumnName> files present in the source project's root directory.

Once a file has been selected, you will need to specify which Movicon.NExT project language will be associated to the imported strings.

## WebHMI Creation Wizard

This wizard helps the developer to create a new project designed for WebHMI Runtime use (INSERIRE UN LINK AL CAPITULO).



You will be asked to define the project name and the destination path to automatically configure the project with the following properties:

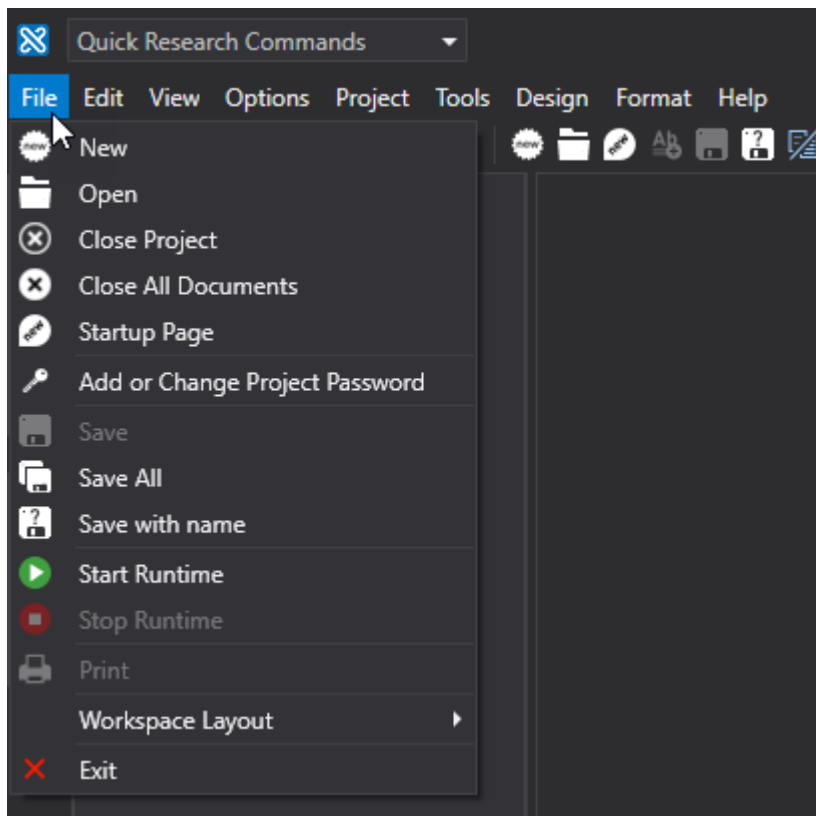
- Project property: "Project Type" set as WebHMI
- I/O Data Server transport set as opc.tcp
- I/O Data Server Database set as SQLite

## 2.3. Menus and Toolbars

The user interface in the development environment has a Menu bar and a Toolbar.

### Menu Bar

The Menu Bar is a common graphical interface pane component and is built with one or more elements that, once activated, display a dropdown menu showing the operations that can be executed by the user.



The main items on the Movicon.NExT menu bar are:

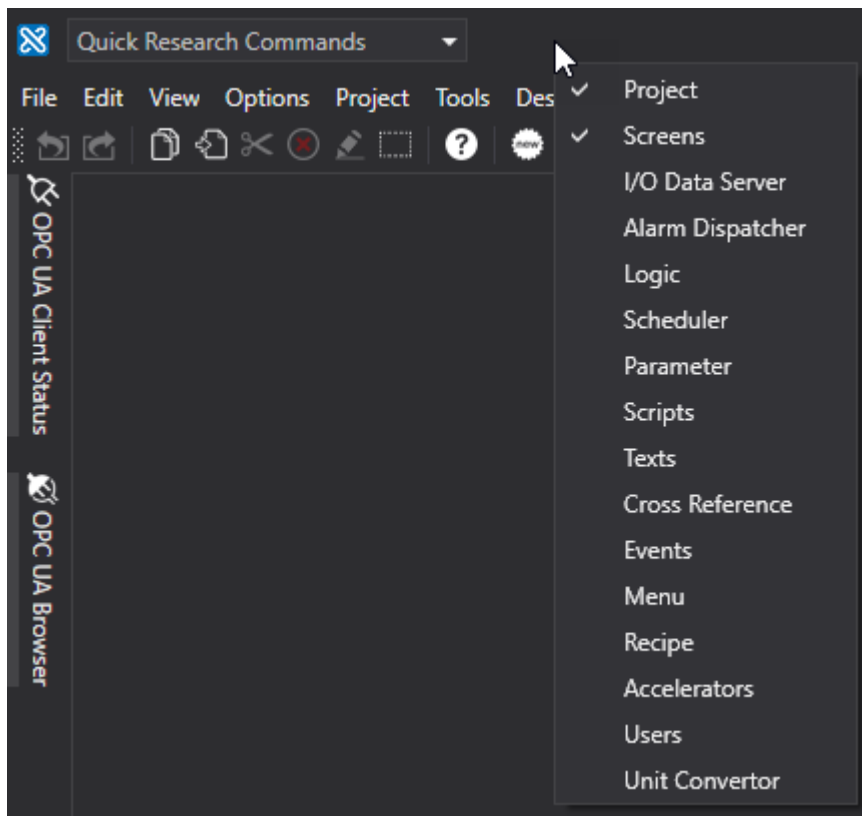
- File: consents users to open, create, save a new project, Start/Stop Runtime and manage screen layouts.
- Edit: gives access to the copy, paste, cut and other commands.
- View: this allows users to call window that compose the layout or to manage active toolbars.
- Options: consents users to manage the Services Control Panel, Web deployment, licenses and other.
- Project: consents users to operate in the project resources.
- Tools: collects all the import/export tools provided in Movicon.NExT.
- Design: consents users to manage the screen zoom and grid visualization.
- Format: consents users to manage object designs on screen such as alignments, visibility and other.
- Help: opens the guidelines.

## Toolbar

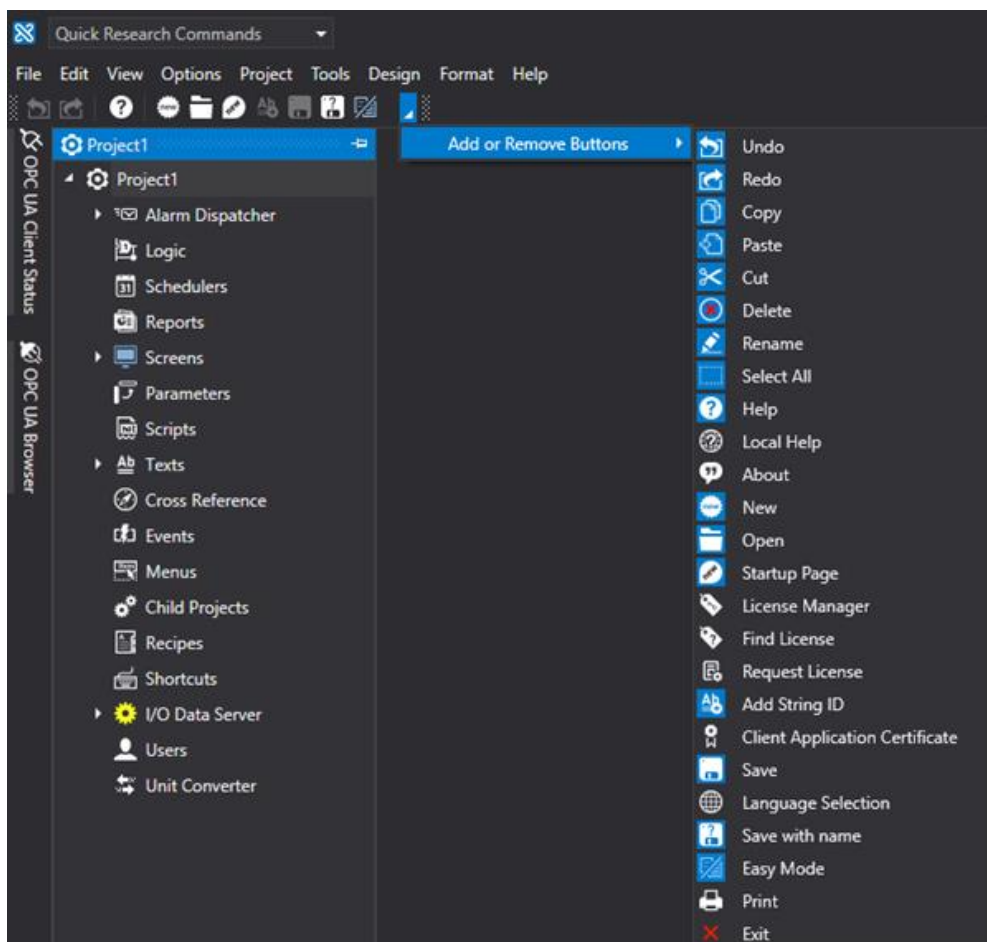
The Toolbar is a horizontal bar with a collection of the most commonly used software functions represented by icons.



Toolbars can be configured and made active by means of "View > Toolbar" menu item or simply by right clicking at the top of the workspace. In the below screen shot, the Project and Screen Toolbars are active along with the "Main Toolbar" which is always visible.



The contents of each Toolbar can also be customized by clicking on the Main Toolbar's customize toolbar icon as shown in the screen shot below.

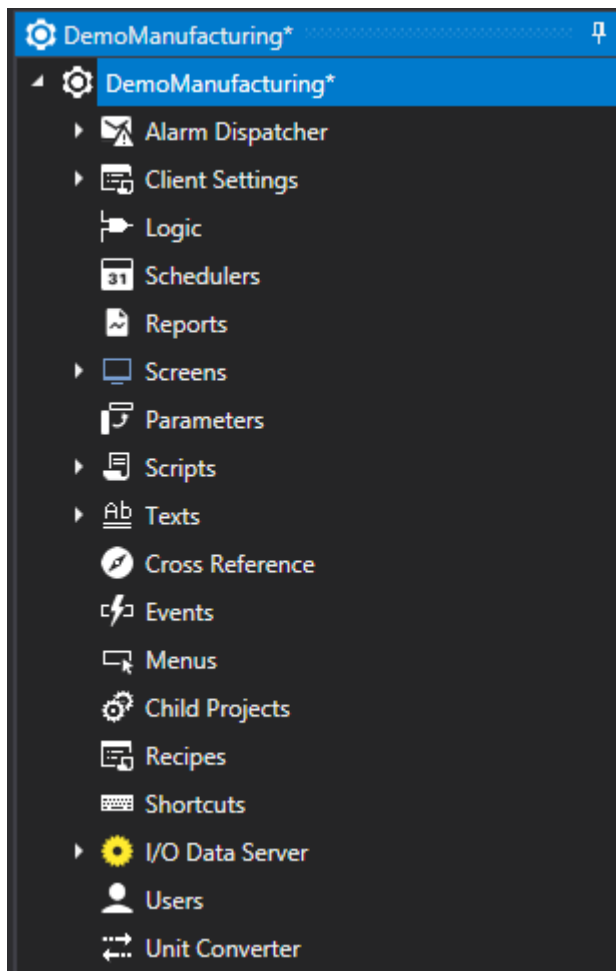


In this way each individual icon on each Toolbar can be activated/deactivated.

## 2.4. Project Explorer Window

The Project Window (or Project Explorer Window) is where all the project resources are listed in a tree structure. This window is used for displaying all information relating to the resources which form the essence of the project itself.

The 'Project Explorer' window gathers all the Resource Groups into one tree structure. When selecting each single Resource Group or a group sub-component, the properties of the object in question will activate and show in the "Property Window" for modifying and setting as necessary.



*Movicon Project Window*

The data structure on disk will be different according to the choice made between XML file or Database for creating a project.

- If you have chosen to use a project on Database, the project's data will be saved coded in a table structure for the database type chosen (eg. Ms SQL Server).
- If you have chosen to create the project on XML file, the system will create a project folder and a series of sub-folders within which the various files containing project information will be saved.

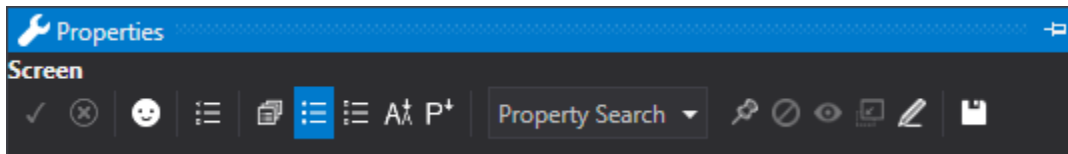
In both cases, however the project's structure will be managed within the workspace area using the main project explorer window which presents the project's modules and resources in a tree structure as shown below.

Each individual module or resource is described in this guide.

## 2.5. Properties Window

The "Property Window" is essential to all projects for setting the characteristics of each PlatformNext resource and component.

Movicon, by virtue of its extreme simplicity, always uses the Properties Window for configuring its resources. The "Properties Window" automatically refreshes each time a project resource or object is selected, by displaying the selected object's inherent properties.



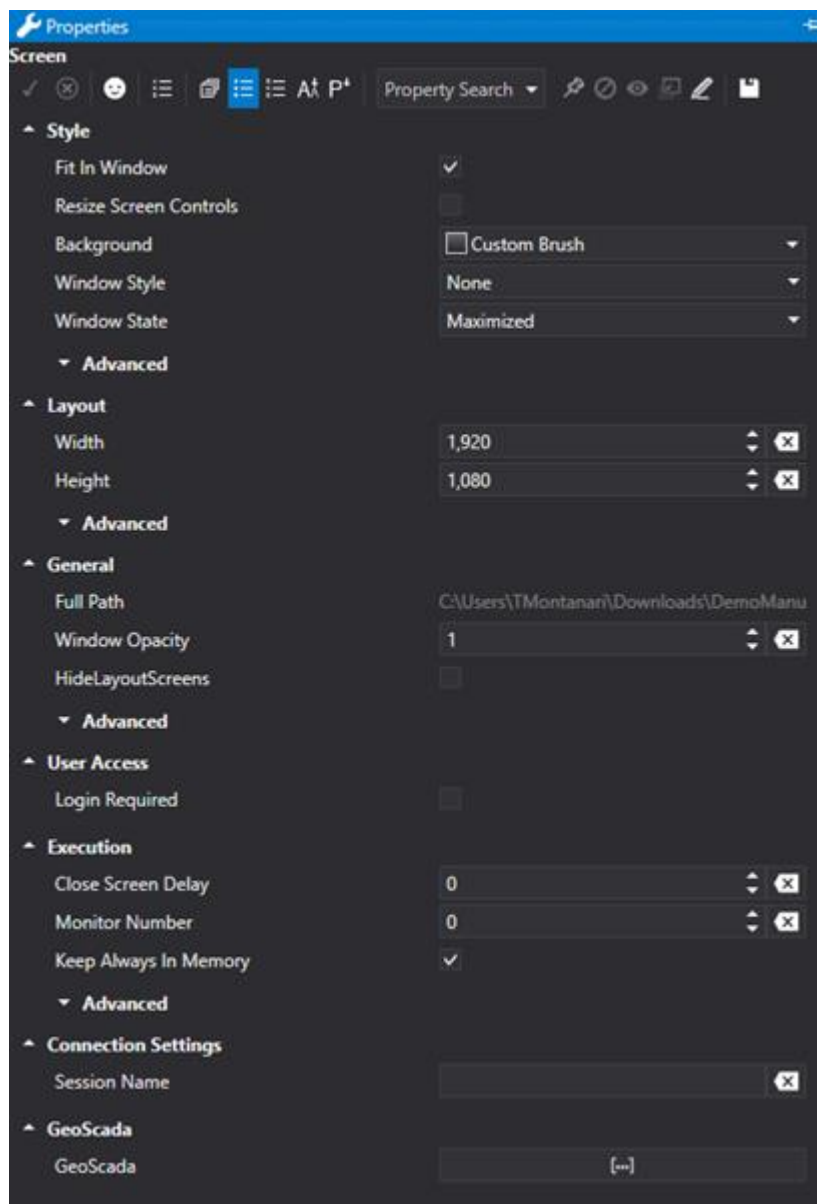
The icons on the Properties Window Toolbar are (from left to right):

- Accept Changes
- Abort Changes
- Easy Mode
- Show changes properties only
- Expand/Collapse Groups
- Group by Category
- Group by Type
- In alphabetical Order
- Order by Priority
- Search

### 2.5.1. Properties of multiple object selections

The property window can also be used to edit a multiple selection of objects simultaneously as shown in the screenshot below:





The title of the window shows as 'Multiple Selection' and all the properties that are not the same are marked with the word "Multiple".

Changes made to properties of a multiple selection will effect all the objects selected.

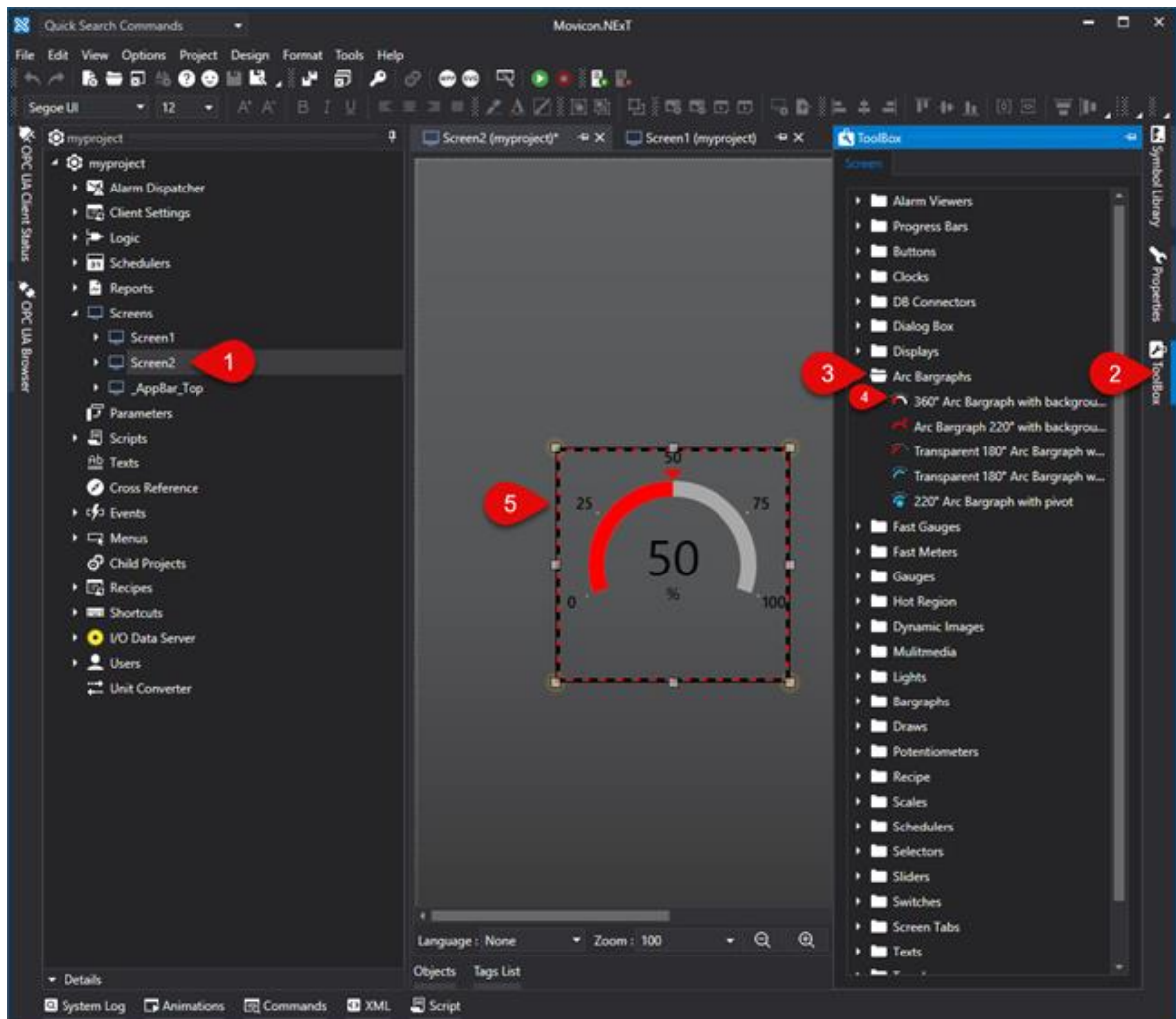
## 2.6. Toolbox Window

### 2.6.1. Toolbox

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



In the following link you will find all the symbols in the Toolbox Library in PDF.  
[https://www.progea.com/wp-content/uploads/2021/11/toolbox-doc\\_raster.pdf](https://www.progea.com/wp-content/uploads/2021/11/toolbox-doc_raster.pdf)



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.

## Inserting objects from the Toolbox

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

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



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



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

**Flour\Egg\nSugar**

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

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

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



Texts in objects that use String ID can be customized using Special string ID.

### 2.6.2. Using Engineering Units

During runtime the objects' value scaling can be defined dynamically using the Engineering Units associated to the Tag variables. When an engineering unit is defined and associated to a Tag, the Min and Max. values of the Tag are automatically scaled and calculated based on the established conversion values during project runtime.

## 2.7. Alarm Viewers

### Alarm Windows

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

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

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



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

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

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

<b>Ack All</b>	This command button is used to acknowledge all the previously selected alarms in the Alarm window.
<b>Reset All</b>	This command button is used to reset all the alarms in the Alarm window without selecting them beforehand.
<b>Ack</b>	This command is used to acknowledge the alarm selected in the Alarm Window.
<b>Reset</b>	This command button is used to reset the alarm selected in the Alarm Window.
<b>Add Comment</b>	This command button is used for entering a comment for the alarm selected in the window. This button is enabled only when one or more alarms have been selected. When pressing the button, a dialog window will appear through which a comment can be entered and associated to the selected alarm.
<b>Shelve</b>	This command button postpones the alarm for a later time which is set using the box located at the side of the button.
<b>UnShelve</b>	This command button is used to cancel the previous command.
<b>Refresh</b>	This command button refreshes the window's contents.
<b>Disable Sound</b>	This command button enables or disables the alarm acoustics associated to the unacknowledged alarm priorities.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

### Border Color

This is used for setting the container's border color.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more languages defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed.

### **Alarm Area Font**

This allows you to set a font for the alarm area.

### **Header Font**

This is used to set the style for the selected font for the object's column header.

## **Layout**

### **Position X**

This is used to set the object's horizontal position in pixels.

### **Position Y**

This is used to set the object's vertical position in pixels.

### **Height**

This is used to change the object's height.

### **Width**

This is used to change the object's width.

### **Border Thickness**

This is used to set the container's border thickness.

### **Margin**

This is used to set the object's margins.

### **Lock Object Position**

This locks the object's position on screen.

## **Visibility**

### **Transparency**

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.



In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

#### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

#### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

#### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

#### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

#### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

#### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

#### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

#### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

#### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

#### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

### **Runtime Configurable**

Enabling this option will make the object configurable at runtime using an appropriate modal window.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

### **Configurable Write Access Level**

This is used to set the minimum access level allowed for editing/configuring the object at runtime.

### **Configurable Write Access Mask**

This is used to set the access area to allow the object to be edited/configured at runtime.

## **Animations**

### **Animations**

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## **Commands**

### **Commands**

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.

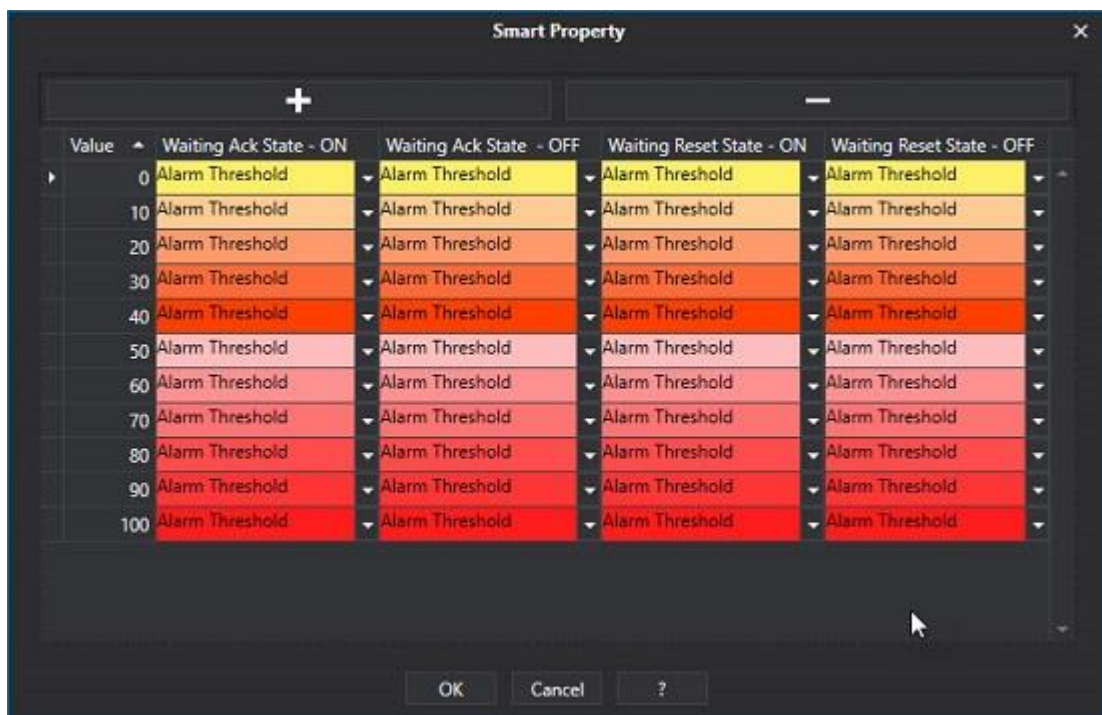


If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Alarm Window

### Object Configuration

A different color for each Severity can be assigned to the different Alarm States in this window.



### Layout Configuration

This is used to modify the object's layout, for instance by adding/removing columns to be displayed at runtime.

### Text Wrapping

This allows you to choose the text wrapping mode of the list of alarms represented in the Alarm Window.

- WrapWithOverflow: the text will go back to the beginning only when the characters reach the maximum space occupied by the column. Cases in which the text refers to one continuous string, this will be broken off when reaching the maximum column width.
- NoWrap: The text will not be wrapped to the beginning and will be broken off.
- Wrap: The text will always be wrapped to the beginning in cases where the characters of one word is too long when reaching the maximum column width, it will be broken off and wrapped down to the next line.

### Button Size

This is used to set the size of the command buttons.

### Button Position

Determines the position of the command buttons.

### Horizontal Column Alignment

This is used to set the horizontal alignment of the window columns.

### Date/Time Format

This is used to set a string to change the object's Date/Time style.

- **Standard format strings for Date/Time values**

The standard Date/Time format strings contain one character. This character defines the model used for representing the value (and in which mode to display the number of years, months and so on).

Format Specifier	Description	Result of formatting a sample value (March 25 2019 12:00:00)
d	Short date pattern	3/25/2019
D	Long date pattern	Monday, March 25, 2019
t	Short time pattern	12:00 AM
T	Long time pattern	12:00:00 AM
f	Full date/time pattern	Monday, March 25, 2003 12:00 AM
F	Full date/time pattern	Monday, March 25, 2003 12:00:00 AM
g	General date/time pattern	3/25/2019 12:00 AM
G	General date/time pattern	3/25/2019 12:00:00 AM

- **Custom format Strings for Date/Time**

To create models for Date/Time values, you will need to combine the strings listed in the table below. These strings represent the year, month, day, number and so on in different formats.

Symbol	Meaning	Result of formatting a sample value (March 25 2019 12:00:00)
yy	The last two digits of the year.	19
yyyy	A four digit year	2019

MM	The number of the month	03
MMM	A short text description of the month	Mar
MMMM	The full name of the month	March
dd	The number of the day	25
ddd	A short text for the day of the week	Mon
dddd	The full name of the day of the week	Monday
/	Date separator. Its appearance depends on regional settings.	

Symbol	Meaning
hh	Hours
mm	Minutes
ss	Seconds
tt	If present represent data in AM/PM format
:	Time separator

- **Custom format Strings for TotalTimeON statistic values**

Symbol	Meaning
hh	Hours
mm	Minutes
ss	Seconds
f	Tenths of Seconds
ff	Hundredths of a second
fff	Milliseconds



You must always use "\" separator character.

Example:

- dd\\.hh\\:mm -> 30.10:20

- hh:mm:ss -> 10:20:52
- mm:ss -> 20:52
- dd\hh:mm:ss\fff -> 30.10:20:52.657



Customized DateTime format settings might not be supported in WebHMI.

### Auto Column Layout

This property is used to enable the "Show Best Fit Button" at runtime. When this property is set to False at runtime, the "Show Best Fit Button" will not be enabled and therefore cannot be used.

### Allow Resizing

This is used to reset the size of the columns at runtime.

### Graphic Buttons

Enable this to replace text in buttons with icons.

### Ack All Button

This enables the ACK ALL button to show to acknowledge all the active alarms.

### Reset Button

This enables the RESET button to show to reset the selected alarm.

### Ack Button

This enables the ACK button to show to acknowledge the selected alarm.

### Reset All Button

This enables the RESET ALL button to show to reset all of the alarms.

### Refresh Button

This enables the REFRESH button to show.

### Shelve Button

This enables the SHELVE button to show to shelve the selected alarm.

### Comment Button

This enables the ADD COMMENT button to show to add a comment to the selected alarm.

### Sound Button

This enables the DISABLE SOUND button to show to disable the selected alarm sound.

### Execute Button

This enables the Execute commands button to show.

### **Use Universal Date and Time (UTC)**

This property allows the SourceTimeStamp to be used for displaying the time in which the alarm was triggered in UTC.

### **Show Server Title**

This enables the server name to be displayed using a label placed in the top bar.

### **Show Quality Status**

This enables the label showing the connection quality status in the top alarm window bar.

### **Show Best Fit Button**

This enables the Best Fit button to be displayed in the top alarm window bar. This button is used to automatically reset the sizes of the columns within the window by adapting in the best way possible at runtime.

### **Show Unshelved Alarm Button**

This enables the Unshelved Alarm Button to show.

### **Show Expand/Collapse Button**

This enables the Expand/Collapse button to show in the top alarm window bar.

### **Show Group Panel**

This enables a dialog window to show through which the alarm list can be grouped based on the headers of columns dragged onto it.

### **Show Search Box**

This enables the search box to show in the top alarm window bar.

### **Show Filter Box**

This enables the Filter box to show on the top alarm window bar.

### **Add/Remove Columns at Runtime**

This allows columns to be added or removed at runtime.

### **Disable Blink Animation**

This is used to disable the alarm blink animation in alarms still to be acknowledged.

### **Severity Required For Comment**

This allows a comment to be added by the user which will be displayed according to the severity level value set in this property. When leaving this property set = 0, no comment will be requested.

### **Connect Child Project Alarms**

This allows child project alarms to display in the same parent project alarm window. This property will display the name of the project that the alarms belong in the "Child Project Name" column. In addition to this setting and in order for this function to be fully optional, you will need to change the "Client Connection Settings" in the parent project's contextual menu, by adding as many items as there are child projects of the alarms you wish to control. Each one should be configured based on the child project's I/O Server app and the name of the host on which the child project is running.

## User Based Runtime Settings

Enabling this option will allow the style of the object to be customized based on user at runtime.



In cases where the Alarm Window object is used by the **"Web Client HTML5"**, the "ACK ALL", "RESET ALL" and "REFRESH" buttons are available.

The "Search" field, column sizes and filter with clauses are not managed.

The Web Client does not currently support alarm comments.

The "Alarms" web page is always present in the web client and can be accessed from the home page. This page allows alarms to be managed on the Web Client to overcome some limitations of the Alarm Window object when deployed on the Web Client: acknowledging and confirming individual alarms, editing filters on alarms.

## Alarm Banners

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

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

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

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

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements"

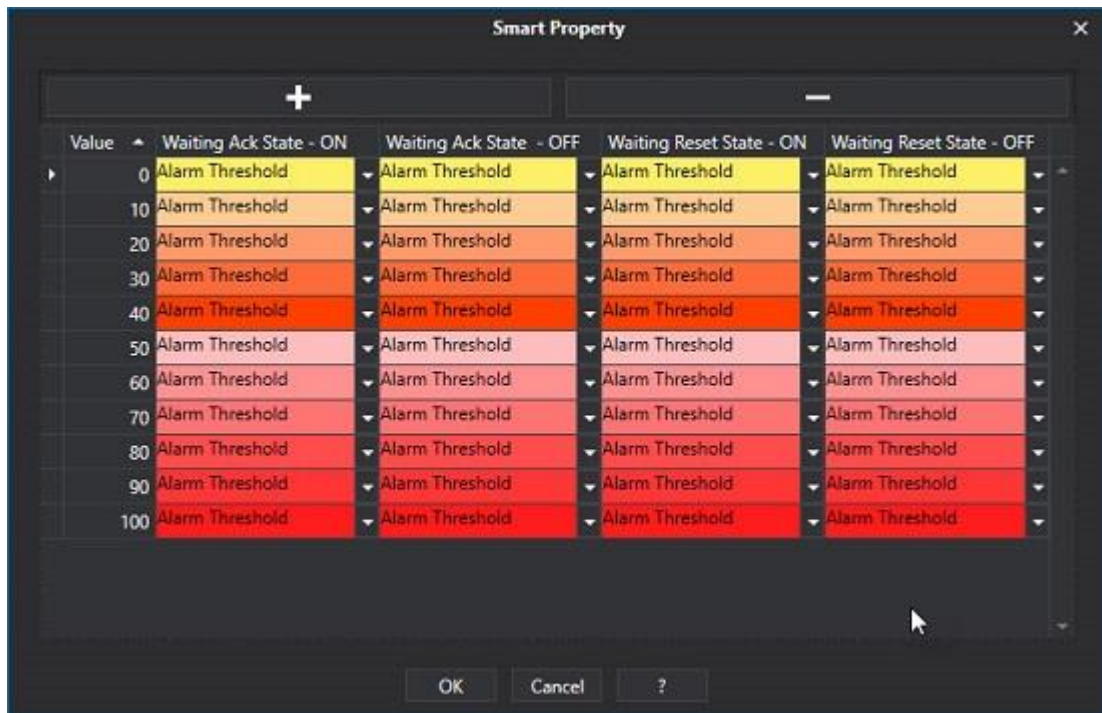


or to define a formula as described in topic: "other expressions"

## Alarm Banner

### Object Configuration

This is used to assign each priority a different color for the different Alarm states within this window.



### Alarm Source

This field allows you to select an Area or a Source from those defined in the project alarms. In this way, the Alarm Banner object will only display the alarms belonging to the Area or Source selected here at Runtime.

This property can also be set dynamically at runtime by using the object's appropriate "AlarmsSource" script property (for further information please see the property's description).



The Dragging&Dropping an Alarm Area or Source onto the object (this command can be executed from the "OPCUA Server Browser" window once the Server has started up) sets the Area or Source in the Alarm Banner's contextual Tag property without changing the "AlarmSource" property. When the Alarm Banner's "AlarmSource" property is left blank, a filter will be performed according to the Area or Source dropped onto the object. On the other hand, when "AlarmSource" property has been defined, this will have priority over the contextual tag field and therefore the filter will be performed according to the "AlarmSource" property.

### Alarm List Default Text

This is used to display the set message when there are no active alarms.

### Scrolling Time

Sets the alarm text scrolling time (available only when the "Auto-Scroll" option has been enabled).

### Spin Position

This is used to change the spin button's position.

#### **Spin Size**

This is used to change the spin button's size.

#### **Show Data and Time**

When this property is set to False the Alarm Banner will only show the alarm text without the relative Date and Time.

#### **Enable Spin**

This is used to enable the spin button.

#### **Show Refresh Button**

Enables the refresh button.

#### **Auto-Scroll**

Enables the alarms to be scrolled automatically.

#### **Disable Blink Animation**

This allows you to disable the blink animation in alarms still to be acknowledged.

## **Style**

#### **Background Color**

This is used to set the container's background color.

#### **Text Color**

This is used to set the object's text color.

#### **Horizontal Content Alignment**

This is used to horizontally align all the contents within the object.

#### **Vertical Content Alignment**

This is used to vertically align all the contents within the object.

#### **Disable Anti-Aliasing**

This is used to Enable/Disable the Anti-aliasing function.

#### **Border Color**

This is used for setting the container's border color.

#### **Spin Background**

This is used to set the color of the Spin button's background

#### **Refresh Button Background**

This is used to set the color of the Refresh Button's background.

## **Fonts**

#### **Font Size**

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### **Height**

This is used to change the object's height.

### **Width**

This is used to change the object's width.

### **Border Thickness**

This is used to set the container's border thickness.

### **Margin**

This is used to set the object's margins.

### **Lock Object Position**

This locks the object's position on screen.

## **Visibility**

### **Transparency**

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

### **Visibility**

This is used to enable the object's visibility.

### **Visible on Web Client**

This makes the object visible on Web Client.

### **Visibility Layer Level**

This is used to set the mask that determines the object's visibility.

### **Zoom Level Visibility X**

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### **Zoom Level Visibility Y**

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## **General**

This is a list of the general properties which are the same for objects and draws:

### **ToolTip**

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Historian Log Viewer

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

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

The events that can be recorded in the Historical Log are:

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

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.



Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

### Border Color

This is used for setting the container's border color.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more languages defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### **TAB Stop**

This makes the object selectable at runtime using the TAB button.

### **Text Direction**

This allows you to set the direction of the text within the object.

### **Enable UI**

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

1. Font Family
2. Font Size
3. Font Style
4. Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

#### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

#### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

#### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

#### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

#### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

#### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

#### **Configurable Write Access Level**

This is used to set the minimum access level allowed for editing/configuring the object at runtime.

#### **Configurable Write Access Mask**

This is used to set the access area to allow the object to be edited/configured at runtime.

## **Animations**

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Database Viewer

### Edit Layout

This allows you to edit the object's graphical layout.

### Filter Event Type

This is used to set the event type for which messages will be loaded and displayed in the window.

### Filter Type

This is used to set the which filter to use according to the period of time wished to be examined.

### Max Rows Nr.

Maximum number or rows (records) to be shown in the window.

### Start Time

This is used to set when to start data sampling.

### End Time

This is used to set when to finish the data sampling.

### Connection String

This is used to set a connection string to the database.

### Command Timeout

This is used to set a DB connection timeout, after which the connection will be disconnected (default 30s).

### User Based Runtime Settings

When enabled, this option permits the object's style to be customized based on the user.

### String List

Shows/Hides the list of strings associated to the object.

## 2.8. Progress Bars

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

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

### **Bargraphs (Evolved)**

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

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

### **Scales**

The Movicon NExT scale objects are configurable and show variable values graphically.

The Scale object has a scale and an indicator which can both be configured using their associated properties.

## **Tag**

### **Tag**

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### **Expression and Reverse Expressions**

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## **Indicator - Basic Aspect**

### **Show Value**

Enable to show the associated numerical value.

### **Indicator Background Color**

This is used to set the indicator's background color.

### **Border Style**

This is used to set the style of the object's borders.

### **Layout Mode**

This is used to set the object's layout mode.

### **Engineering Unit**

This is used to set the name of the engineering unit. For further information, please refer to topic on "Using Engineering Units"



### Use Engineering Unit Limits

When enabling this property, the object will use the limits defined in the Engineering Unit linked to the Tag and will ignore the static limits it was set with. Otherwise, the limits used will be those given by the most restrictive minimum and maximum limits between the Static Limits and the Engineering Unit associated to the Tag.

This property will be ignored when the object has been set with Dynamic Limits.

In addition, when this property is enabled, the object will also inherit the Unit of Measure set in the Engineering Unit linked to the Tag. When the Unit of Measure has also been set in that object, this will have priority over the one set in the Engineering Unit.

For further information, please see "Using Engineering Units" .

### Value Offset

This is used to set the offset for the numerical value's position.

### Numerical Value Format

This is used to set the value's numerical format (e.g. 000.0).

## Indicator - Marker

### Show Marker

Shows/Hides the marker.

### Marker Color

This is used to set the color of the marker.

### Marker Border Color

This is used to set the marker's border color.

### Marker Height

This is used to set the marker's height.

### Marker Width

This is used to set the marker's width.

### Marker Offset

This is used to set the distance of the marker's distance from the border.

### Marker Animation

This is used to enable marker animation.

### Clickable Marker

This is used to enable the marker's interactivity so that the user can set the desired value by clicking on the marker and move it to the desired value.

### Marker Orientation

This is used to set the marker's orientation within the object.

## Indicator - Dividers

### Tickmark Style

This is used to set the object's tickmark style.

### Major Tickmark Fill

This is used to set the color of the major tickmarks.

### Major Interval Count

This is used to set the number of major scale tickmarks.

### Minor Tickmark Fill

This is used to set the color of the minor tickmarks.

### Minor Interval Count

This is used to set the number of minor scale tickmarks.

### Major Tickmark Length

This is used to set the major tickmarks on the object's scale.

### Major Tickmark Thickness

This is used to set the thickness of the major tickmarks.

### Major Tickmark Offset

This is used to set the distance of the major tickmarks from the object's border.

### Show First Major Tickmark

Shows/Hides the scale's first major tickmark.

### Show Last Major Tickmark

Shows/Hides the scale's last major tickmark.

### Minor Tickmark Length

This is used to set the length of the minor tickmarks.

### Minor Tickmark Offset

This is used to set the distance of the minor tickmarks from the object's border.

### Minor Tickmark Thickness

This is used to set the thickness of the minor tickmarks.

### Show Minor Tickmarks on Major

This enables the minor tickmarks to show in the same position of a major tickmark.

## Indicator - Range Bar

### Show Range Bar

Enables/disables the Value Range Bar.

### Range Bar Color

This is used to set the color of the Value Range Bar.

### Range Bar Background

This is used to set the background color of the Range bar.

### Range Bar Thickness

This is used to set the Range Bar's thickness.

### Range Bar Layout Index

This is used to set the Range Bar's depth index.

### Range Bar Offset

This is used to set the space between the Range Bar segments.

### Range Bar Animation

Enables/disables the Range Bar animation. When enabled, the bar will fill until it reaches the relative position during animation.

### Range Bar Clickable

Enables/disables interactivity with the Range Bar.

## Indicator - Value Bar

### Show Value Bar

Enables/disable the Value Bar.

### Value Bar Color

This is used to set the Value Bar's color.

### Value Bar Background Color

This is used to set the Value Bar's background color.

### Value Bar Thickness

This is used to set the Value Bar's thickness.

### **Value Bar Offset**

This is used to set the distance between the Value Bar segments.

### **Value Bar Animation**

Enables/disables the Value Bar animation. When enabled, the bar will fill until it reaches the relative position during animation.

### **Value Bar Clickable**

Enables/disables interactivity with the Value Bar.

## **Indicator - Labels**

### **Label Color**

This is used to set the color of the scale's labels.

### **Value Label Color**

This is used to set the color of the numerical value on the label.

### **Engineering Unit Label Color**

This is used to set the color of the value relating to the engineering unit.

### **Engineering Unit Label Offset**

This is used to set the space between the text and borders of the label.

### **Label Orientation**

This is used to set the orientation of the scale's labels.

### **Label Offset**

This is used to set the distance between the scale's labels from the border.

### **Label String Format**

This is used to set the labels' text format.

### **Show First Label**

Enabling this will show the first label.

### **Show Last Label**

Enabling this will show the last label.

### **Show Engineering Unit Label**

Enabling this will show the engineering unit label.

### **Auto Hide Elements**

When this property is set to true, the elements will gradually disappear and the number of ticks will lessen as the control is made smaller and as a consequence so will the labels. For example, as the object is gradually made smaller, the numerical values will start to disappear in the order they are shown, along with the ticks, the scale numeration, the indicator and the border and so forth.

When the property is set to False, all the elements will remain visible when the object is reduced in size but may overlap each other as a consequence.



This behaviour is determined by the fact that when re-sizing the object, the elements, such as the scale and fonts, do not re-size accordingly but remain their initial set sizes.

If you wish to resize the object so that all its elements re-size accordingly, you will need create a symbol with the object's initial sizes, and then when you re-size it (so that it becomes a ViewBox) all its elements will re-size proportionately.

## Indicator - Alarm Zone

### Show Alarm Zone

Enables the corresponding Scale Zone (1,2 or 3) to show.

### Scale Zone Color

This is used to set the color of the scale zone.

### Alarm Scale Zone Start

This is used to set the scale zone's start position (zone 1, 2, 3).

### Alarm Scale Zone End

This is used to set the scale zone's end position (zone 1, 2, 3).

### Scale Zone Thickness

This is used to set the scale zone's thickness (zone 1, 2, 3).

### Scale Zone Offset

This is used to set the offset between the scale zone segments (zone 1, 2, 3).

### Scale Zone Layout Index

This is used to set the scale zone's depth.

## Indicator - Scale

### Minimum Value

This is used to set the allowed minimum value.

### Maximum Value

This is used to set the allowed maximum value.

### Tag Min. Value

This replaces the minimum scale value with the linked variable value.

### Tag Max Value

This replace the maximum scale value with the linked variable value.

## Value Entry Limits for Controls

The EditDisplay, Gauge, LinearMeter, Slider and Potentiometer control types allow tag values to be displayed and changed. When entering values, you will be able to define the Static Limits (defined in Design mode) or Dynamic Limits (linking two runtime editable tags). In addition, these limits can also be defined in the same Tag by means of the link to an Engineering Unit Tag. The priorities with which these limits are used to intervene depend on certain settings. Below are listed the cases in which these limits may intervene:

4. The Tag is not linked to an Engineering Unit. In this situation, only the Static Limits or the Dynamic Limits are considered. The Dynamic Limits always have priority over Static Limits, therefore, if Tags have been insert to manage the Dynamic Limits in the object, these will be used instead of the Static Limits at Runtime.
5. The Tag is linked to an Engineering Unit and has been set with Static Limits and not Dynamic Limits. In this case, the control's "Enable Engineering Unit" property comes into play. When the property is enabled, the control will inherit the Tag's Engineering Unit's limits and will only use those. Conversely, if the control's "Enable Engineering Unit" property is disabled, the most restrictive limits (both the minimum and maximum) will be considered between the Engineering Units and the Static Limits. For instance, supposing that the Engineering Unit limits are min=0 and max=100 and that the control's Static Limits are min=10 and max=200, the control will then use min=10 and max=100 as limits (the maximum of the minimums and the minimum of the maximums).
6. The Tag has been linked with a Engineering Unit and set with Dynamic Limits. In this case, the most restrictive limits will always be considered (both minimum and maximum) between the Engineering Unit and the Dynamic Limits. For instance, supposing the Engineering Units are min=10 and max=100 and that the control's Dynamic Limits are min=5 and max=90, the control will then use
7. min=10 and max=90 as limits (the maximum of the minimums and the minimum of the maximums).

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

### **Label Font**

This is used to set the font to use in the labels.

### **Label Value Font**

This is used to set the font of the label's actual value.

### **Engineering Unit Font**

This is used to set the engineering unit label's font.

## **Layout**

### **Position X**

This is used to set the object's horizontal position in pixels.

### **Position Y**

This is used to set the object's vertical position in pixels.

### **Height**

This is used to change the object's height.

### **Width**

This is used to change the object's width.

### **Border Thickness**

This is used to set the container's border thickness.

### **Margin**

This is used to set the object's margins.

### **Lock Object Position**

This locks the object's position on screen.

## **Visibility**

### **Transparency**

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



### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### Force as Dynamic on Client

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

#### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

#### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

#### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

#### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

#### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

#### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

#### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

#### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

#### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

#### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## User Access

### Access Level

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### User Access Role

This is used to set the name of the user password group to allow user interaction with this object.

### Access Level from Tag

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## 2.9. Buttons

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



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

### Buttons-Commands

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

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

### Tag

#### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

#### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

### Button Settings

#### Click Mode

This is used to set the click mode to execute the command with (release, press, hover).

#### Pressed Icon

This is used to set an image to display on the button when in the pressed state.

#### Released Icon

This is used to set an image to display on the button when in the released state.

#### Checked Icon

This is used to set an image to display on the button when in the 'selected' state. This property is only considered for Toggle Button types such as the CheckBox.

### Disabled Icon

This is used to set the image to display when the button is in the disabled state and therefore not clickable

### Overlap Text on Image

When enabled, the button's text will display overlapping the image, otherwise the text will display at the side of the image, each occupying half of the button.

### Stretch Image

This is used to set how the image adapts to button:

- *None*: no adaption
- *Fill*: the image will adapt to the size of the button. In this case the sizes of the image will adapt to those of the button and the proportions might not be respected.
- *Uniform*: the image will be adapted to the size of the button by displaying the whole image and keeping it to proportion. In this case the image might not cover the whole surface of the button but will be in proportion.
- *Uniform toFill*: the image will be adapted to the size of the button keeping to the image's proportions. In this case the image will always adapt to one of its sizes (height or width) but it might be cut partly off either end.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### **Height**

This is used to change the object's height.

### **Width**

This is used to change the object's width.

### **Border Thickness**

This is used to set the container's border thickness.

### **Margin**

This is used to set the object's margins.

### **Lock Object Position**

This locks the object's position on screen.

## **Visibility**

### **Transparency**

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

### **Visibility**

This is used to enable the object's visibility.

### **Visible on Web Client**

This makes the object visible on Web Client.

### **Visibility Layer Level**

This is used to set the mask that determines the object's visibility.

### **Zoom Level Visibility X**

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### **Zoom Level Visibility Y**

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## **General**

This is a list of the general properties which are the same for objects and draws:

### **ToolTip**



This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### Force as Dynamic on Client

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

### Preserve Size

This is used to preserve the values of the local size property group when the object is updated from the library.

### Preserve Commands

This is used to preserve the list of commands when the object is updated from the library.

#### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

#### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

#### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

#### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

#### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

#### **Configurable Write Access Level**

This is used to set the minimum access level allowed for editing/configuring the object at runtime.

#### **Configurable Write Access Mask**

This is used to set the access area to allow the object to be edited/configured at runtime.

## **Animations**

#### **Animations**

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## **Commands**

#### **Commands**

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Clocks

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

### Tag

#### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

#### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

### Clock Style - Dial

#### Enable Background

This is used to enable/disable the clock object's background.

#### Use System Time Zone

When enabled, the clock object will use the PC's time zone.

#### Clock Time Zone

This allows you to set a custom time zone.

#### Show Clock Text

This enables the current time to be shown in digital format within the analogical clock.

#### Clock Style

This is used to set the style of the clock object.

### Show Hour Label

This is used to show the hour label on the clock face.

### Label Orientation

This is used to set the label orientation type.

### Seven Segment Format

Enabling this will display the time with a seven segment font which cannot be changed.

## Clock Style - Hands

### Hour - Minute - Second hand Style

These properties are used to set the styles of the hour, minute and second hands.

### Handle Z-index

This is used to set the hand object's depth level.

### Show Hour - Minute - Second hand

These properties are used to hide/show the hour, minute and seconds hands.

### Enable hand animation

This is used to activate/deactivate the hand animation.

## Clock Style - Pin

### Pin Cap Style

This is used to set the clock object's pin cap style.

### Pin Height

This is used to set the pin's height.

### Pin Width

This is used to set the pin's width

### Pin Cap Z-Index

This is used to set the pin cap's overlapping level.

## Clock Style - Tickmark

### Tickmark Style

This is used to set the style of the clock object's tickmarks.

### **Major Tickmark Length**

This is used to set the length of the clock object's major tickmarks.

### **Major Tickmark Z-Index**

This is used to set the major tickmark depth on the clock object's scale.

### **Major Tickmark Thickness**

This is used to set the thickness of the major tickmarks on the clock object's scale.

### **Major Tickmark Offset**

This is used to set the distance between the major tickmarks and the clock object's border.

### **Show First Major Tickmark**

This is used to show/hide the first major tickmark on the clock object's scale.

### **Show Last Major Tickmark**

This is used to show/hide the last major tickmark on the clock object's scale.

### **Minor Tickmark Length**

This is used to set the length of the clock object's minor tickmarks.

### **Minor Tickmark Z-Index**

This is used to set the minor tickmark depth on the clock object's scale.

### **Minor Tickmark Thickness**

This is used to set the thickness of the minor tickmarks on the clock object's scale.

### **Minor Tickmark Offset**

This is used to set the distance between the minor tickmarks and the clock object's border.

### **Show Minor Tickmarks on Major**

This is used to show/hide minor tickmarks on major tickmarks.

## **Style**

### **Background Color**

This is used to set the container's background color.

### **Text Color**

This is used to set the object's text color.

### **Horizontal Content Alignment**

This is used to horizontally align all the contents within the object.

### **Vertical Content Alignment**

This is used to vertically align all the contents within the object.

### **Disable Anti-Aliasing**

This is used to Enable/Disable the Anti-aliasing function.

### **Clock Text Color**

This is used to set the color of the time in text format.

### **Clock Text Background Color**

This is used to set the background color of the time in text format.

### **Clock Background Color**

This is used to set the clock's background color.

### **Label Text Color**

This is used to set the label's text color.

### **Hour - Minute - Second Hand Color**

These properties are used to set the respective hand colors.

## **Fonts**

### **Font Size**

This is used to set the size of the selected font.

### **Font Family**

This is used to set a font type to use for text.

### **Font Style**

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### **Font Weight**

This is used to select the thickness of the selected font.

### **Set Font based on Language**

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more languages defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

### Font Orario

permette di impostare lo stile dell'orario attuale in formato testuale.

### Font Tacche Orologio

Permette di impostare lo stile del carattere delle etichette.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.



### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### **Menu with Left button**

This enables the contextual menu to show by also using the left mouse key.

### **TAB Stop**

This makes the object selectable at runtime using the TAB button.

### **Text Direction**

This allows you to set the direction of the text within the object.

### **Enable UI**

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

### Preserve Size

This is used to preserve the values of the local size property group when the object is updated from the library.

### Preserve Commands

This is used to preserve the list of commands when the object is updated from the library.

### Preserve Animations

This is used to preserve the list of animations when the object is updated from the library.

### Preserve Menu

This is used to preserve the set contextual menu, when the object is updated from the library.

### Preserve Expression

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## **Animations**

### **Animations**

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## 2.10. DB Connectors

### DB Grid Control

The "DB Connector" Grid object has been designed to create a connection to a table created in a database even in those cases when it is not managed by the Movicon.NExT resources.

The Database connection can be created with a query using the smart properties.

The grid will display data from the table at runtime.

The Grid Control has other smart properties such as: **"Allow Add New"**, **"Allow Remove"**, **"Allow Edit"**.

These properties allow you to add a new recording by adding a row to the grid, remove a row and its relative record from the DB table or edit the value of one or more grid cells.



The Grid object can also be used to display the result of a view.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## DB Connector Grid

**Minimum Row Height**

This is used to set the minimum row height.

**Inhibit cell selection**

When enabled, allows the entire row to be selected to inhibit cell editing.

**Show Cell Border**

This is used to display/hide the border of the rows that compose the Grid.

**Graphic Buttons**

Enabling this will display the object's buttons with icons instead of text.

**Allow Edit**

This is used to enable the editing of the object at runtime.

**Allow Add New**

This is used to enable the command that adds a new row to the grid.

**Allow Remove**

This is used to enable the command that removes selected rows from the grid.

**Max. Transactions**

This is used to enter the maximum transactions to be used for each recording.

**Object Configuration**

See chapter below on Smart Properties

**Layout Configuration**

This is used to configure the object's layout, such as for example adding/removing columns to be displayed at runtime.

**Allow Primary Key Changes**

This allows the value of the primary key column to be edited.

**Show Best Fit Button**

This enables the button in the Grid object's top bar that is used to automatically adapt the layout of the columns so they fit within the object.

**Show Group Box**

This is used to show/hide the group box.

**Show Search Box**

This is used to show/hide the search box.

**Show Filter Box**

This is used to show/hide the filter box.

### Show Command Buttons

This is used to show/hide the command buttons.

### User Based Runtime Settings

Enabling this option will allow the object's style to be customized based on user at runtime.

### Threshold Images

Enabling this will show the various images set for each threshold using the Smart Properties.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

### Column Background Color

This is used to set the Grid columns' background color.

### Column Text Color

This is used to set the text color of the Grid's columns.

### Row Text Color

This is used to set the text color of the Grid's rows.

### Selected Row Background Color

This is used to set the background color of the selected row/s.

### Selected Row Text Color

This is used to set the text color of the selected row/s.

### Cell Border Color

This is used to set the color of the border of the rows composing the Grid.

### Selected Cell Background Color

This is used to set the background color of the selected cell/s.

### Selected Cell Text Color

This is used to set the selected cell/cells' text color.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed.



### Row Area Font Settings

This is used to set the font for the columns' data areas.

### Header Font Settings

This is used to set the font type for the column headers.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### **Is Interactive**

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### **Menu with Left button**

This enables the contextual menu to show by also using the left mouse key.

### **TAB Stop**

This makes the object selectable at runtime using the TAB button.

### **Text Direction**

This allows you to set the direction of the text within the object.

### **Enable UI**

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

### Preserve Size

This is used to preserve the values of the local size property group when the object is updated from the library.

### Preserve Commands

This is used to preserve the list of commands when the object is updated from the library.

### Preserve Animations

This is used to preserve the list of animations when the object is updated from the library.

### Preserve Menu

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

### **Configurable at Runtime**

Enabling this will allow the object to be configurable at runtime with an appropriate modal window.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

### **Configurable Write Access Level**

This is used to set the minimum access level allowed for editing/configuring the object at runtime.

### Configurable Write Access Mask

This is used to set the access area to allow the object to be edited/configured at runtime.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

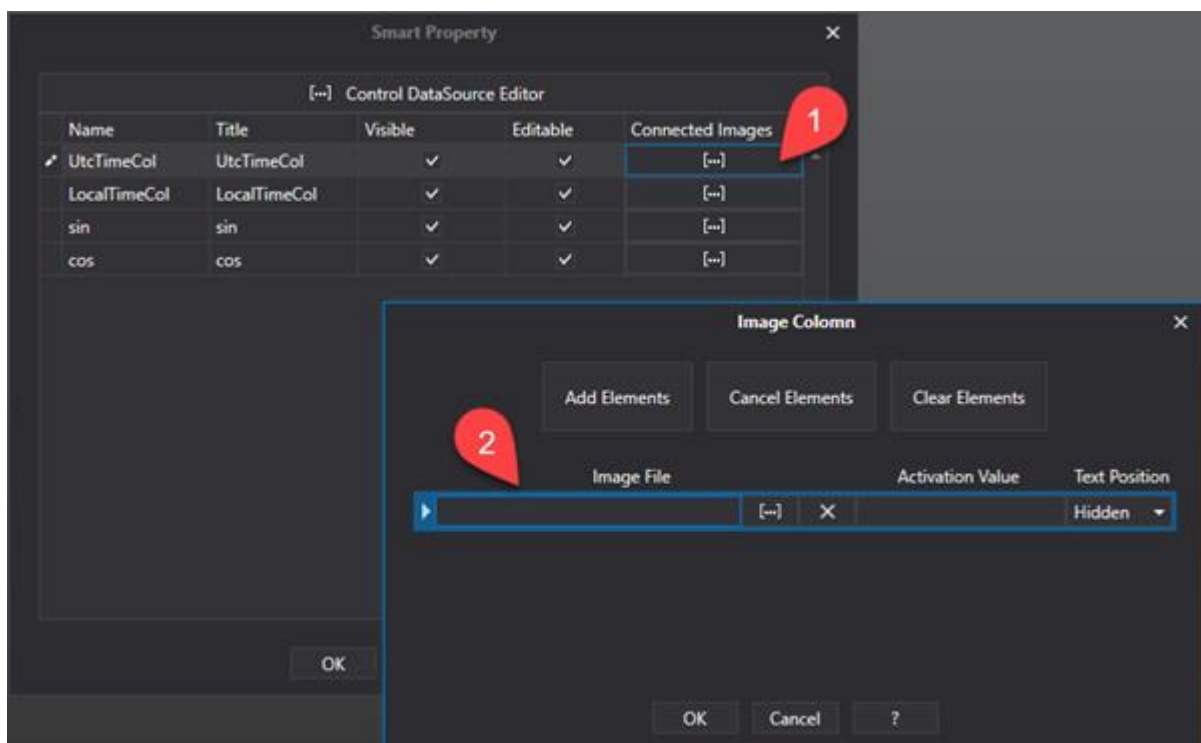
This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Smart Properties (object configuration)



### Control Data Source Editor

Allows you to define the data source and query.



The database must be accessible while configuring the connection string and data extraction query to obtain the grid structure with the necessary columns, otherwise the data will not be displayed.



Please pay extra attention to the fact that the string and query are case sensitive in WebHMI projects running on Linux OS.

### Name

Name of the column resulting from the query.

### Title

Title of the column shown in the object.

### Visible

Shows the columns resulting from the query.

### Editable

Makes each column row editable.

### Connected Images

Allows you to add images to display for each threshold and which can be set using the smart properties popup.

-

## Combo Box

This object comes from the "Dialog Box" category and can be configured by inserting a list of strings separated by the "|" character, or by setting a query to retrieve data from a DB table. Data is not entered but selected from the drop down menu which opens when clicking the arrow.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Combo Box

### Source Reader Settings

This editor allows you to populate the Combo Box with the items from a DB column.

### Item List

This allows you to add combo box items using a string. Each item must be separated by the 'Pipe' character e.g. 1|2|3

### Tag Item Source

This is used to select the variable from which to obtain the displayed source data

### Text

This is used to set a predefined text to show within the text box.

### Selected Index

This is used to set the index value for the first item on the list.

### Toggle Button Size

This is used to set the size of Toggle Button to open list.

### Read Only

When enabled, this property forces the Combo Box as a read only object and therefore the user will not be able to change the value within it.

### Preserve User Input

When enabled, allows the value entered by the user to be used in cases of matching items on a drop down list.

### Allow Editing

When enabling this property, it will be possible to edit the Combo Box value manually at runtime.

### Keep Open on Edit

Enable this so that the drop-down window stays open while editing the value within the text box.

### Keep Items Synchronized

This allows the various item to stay synchronized with their respective sources.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one



language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled by default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### Force as Dynamic on Client

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## User Access

### Access Level

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### User Access Role

This is used to set the name of the user password group to allow user interaction with this object.

### Access Level from Tag

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## DB List Box

The Movicon NExT List Box objects are components that can be inserted anywhere on screen in order to display lists of items. A numeric or string type variable can be associated to this object. Each row of the list corresponds to a numeric value that is made available in the variable associated to the object if this variable is numeric type. If the variable is string type, it will be updated with the selected Item's text. Each time a row is selected, the List Box's variable is updated with the corresponding numeric value (the value is progressive, starting from 0 to indicate the first row) or with the selected Item's text.

### Tag

#### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

#### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## ComboBox

#### Source Reader Setting

This editor allows you to populate the Combo Box with the items from a DB column.

#### Item List

This is used to add combo box items using a string. Each item must be separated by the "Pipe" character, eg. "1|2|3"

#### Selected Index

This is used to set the index value for the first listed item.

#### Selection Mode

This is used to set the selection mode type of items on the list.

#### Keep Items Synchronized

This keeps the selected item synchronized with the relative source.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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



### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

#### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

#### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

#### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

#### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

#### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

#### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

#### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

#### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

#### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

#### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## User Access

### Access Level

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### User Access Role

This is used to set the name of the user password group to allow user interaction with this object.

### Access Level from Tag

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## DB Pivot Grid

The Pivot Grid object is used for creating a pivot table to display multi-dimensional data analysis. The Pivot Grid allows large amounts of data to be summarized and represented in a table where it can then be sorted, grouped and filtered. Users can personalize the table layout according to the type of analysis they wish to perform. This is simply done by dragging & dropping the specific items within the table as desired.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Pivot Grid

### Object Configuration

See chapter below on Smart Properties

### Layout Configuration

This is used to configure the object's layout, such as for example adding/removing columns to be displayed at runtime.

### Show Grand Total Row

Enables the grand total row to show.

### Show Total Row

Enables the total row to show.

### Show Header Row

Enables the header row to show.

### Show Grand Total Column

Enables the Grand Total Column to show.

### Show Grand Total Column Header

Enables the Grand Total Column Header to show.

### Show Total Column

Enables the total column to show.

### Show Header Column

Enables the Header Column to show.

### Show Column Headers

Enables the column headers to show.

### Show Filter Headers

Enables the filter headers to show.

### Show Totals for single values

Enables the totals for single values to show.

### Show Prefilter Panel

Enables the prefilter panel to show.

### Graphic Buttons

Enable to use icons in toolbar.

### User Based Runtime Settings

Enabling this option will allow the object's style to be customized based on user at runtime.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

### Row Area Font

This is used to set the font for the row's area.

### Header Font

This is used to set the font for the column header.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

#### **Position Y**

This is used to set the object's vertical position in pixels.

#### **Height**

This is used to change the object's height.

#### **Width**

This is used to change the object's width.

#### **Border Thickness**

This is used to set the container's border thickness.

#### **Margin**

This is used to set the object's margins.

#### **Lock Object Position**

This locks the object's position on screen.

## **Visibility**

#### **Transparency**

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

#### **Visibility**

This is used to enable the object's visibility.

#### **Visible on Web Client**

This makes the object visible on Web Client.

#### **Visibility Layer Level**

This is used to set the mask that determines the object's visibility.

#### **Zoom Level Visibility X**

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

#### **Zoom Level Visibility Y**

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## **General**



This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be

executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### Force as Dynamic on Client

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

### Preserve Size

This is used to preserve the values of the local size property group when the object is updated from the library.

#### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

#### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

#### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

#### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

#### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

#### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

#### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

#### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

#### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

### Configurable Write Access Level

This is used to set the minimum access level allowed for editing/configuring the object at runtime.

### Configurable Write Access Mask

This is used to set the access area to allow the object to be edited/configured at runtime.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Smart Properties

### Name

Name of the resulting column from the query.

### Title

Title of the column displayed in the object.

### Visible

This allows the resulting column from the query to be displayed.

## 2.11. Dialog Boxe

### Switch Buttons, Lights and Checkboxes

The Movicon NExT buttons are divided into two types according to their mechanics but they all have the same style properties. Some buttons have ON-OFF mechanics while others execute COMMANDS. Therefore you must insert the button type that is most appropriate for the functionality needed.



Be reminded that the styles of each object you insert on screen can be modified graphically but their mechanics cannot.

### Buttons-Digital Checkboxes

In Toolbox section you will find a vast range of buttons by type and style. The main characteristic of this category is the possibility to associate a variable for toggling between 0 and 1 (On-Off command). Even though this object's styles are the same as those of the command buttons, its "Check-Box" mechanics require you to associate it a Tag variable. This can be done by using the Drag & Drop techniques or by associating the Tag using the properties or quick select menu.

### CheckBox

The checkBox is a classical two state selector button. A Movicon NExT variable can be associated to this control and set to the "0" value (unchecked box) or the "1" value (checked box) according to the state of the box.

The state can be selected with a mouse click or using the keyboard to select the by selecting the component with the TAB or MAIUSC+TAB keys and pressing ENTER.

### Lamps

Lamps are luminous indicators that are used to represent the state of the connected variable. This is done by using the behaviour properties to set them to display based on two or three state mechanics.

These objects are read only for default but it is also possible to control the assigned variables through their associated properties.

### Toggle Switches

The MoviconNExT Switches and Selectors are used to execute toggle commands or set project variables. These elements are accessible through the "Buttons Switches" group in the toolbox and a major part of their functions and properties resemble those belonging to the checkboxes. As with the objects from the "Buttons-Checkbox" group, they can be set with tri-state mechanics to assume three different positions. The tri-state mechanics enable the set variable to be modified to assume a new value each time the switch or selector area is clicked on. Therefore with each click the variable passes from the zero value to value one then value two and then back to the zero value.

The switch/selector's image changes in function with the new variable value, alternating between three possible images: **switch/selector in zero position**, **switch/selector in position one** and **switch/selector in position two**. The variable can also be set by keeping the mouse button pressed down on the selector and moving it to another position. In this case the variable will be set to the new value only when the mouse button is released and the switch/selector is in a different position from the one it started with.



To exploit the tri-state animation to the full, use the switches from the "Selector 3P" sub class which are graphically preset for this type of use.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Button Settings

### Click Mode

This is used to set the click mode to generate the command execution (release, press, hover).

### Is Three State

If provided, this option is used to set the button as three-state (only available for Checkboxes).

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

### Back Color When ON

This is used to set the object's back color which will change when object is active.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

#### **Position Y**

This is used to set the object's vertical position in pixels.

#### **Height**

This is used to change the object's height.

#### **Width**

This is used to change the object's width.

#### **Border Thickness**

This is used to set the container's border thickness.

#### **Margin**

This is used to set the object's margins.

#### **Lock Object Position**

This locks the object's position on screen.

## **Visibility**

#### **Transparency**

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

#### **Visibility**

This is used to enable the object's visibility.

#### **Visible on Web Client**

This makes the object visible on Web Client.

#### **Visibility Layer Level**

This is used to set the mask that determines the object's visibility.

#### **Zoom Level Visibility X**

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

#### **Zoom Level Visibility Y**

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## **General**



This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be

executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### Force as Dynamic on Client

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

### Preserve Size

This is used to preserve the values of the local size property group when the object is updated from the library.

#### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

#### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

#### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

#### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

#### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

#### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

#### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

#### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

#### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Check Box Array

The Check Box Array is a set of classic two state selection buttons and which distinguishes itself from the ordinary "Check Box" by having the possibility to associate various check boxes to the one same control. Therefore, a Movicon.NExT variable can be associated which will set to "0" (unchecked box) or "1" (checked box) according the box's status.

The status is selected with a mouse click or by using the TAB or MAIUSC+TAB keys on the keyboard and pressing ENTER.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Check Box

### Orientation

This is used to orientate the various Check Boxes within the control.

### No. of Check Boxes

This is used to set the number of Check boxes to display.

### Check Box Text

This is used to customize the text associated to all the check boxes within the control.

### Check Box Size

This is used to set the size of the various checkboxes within the control.

### Check Box Index Format

This is used to define the which format to show the index of each checkbox with.

### Check Box Margins

This is used to define the check box margins.

### Check Box Wrapping

Activating this property will adapt the checkboxes to the space available within the control.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

### Border Color

This is used to select a color for the check box borders.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled by default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.



### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### Force as Dynamic on Client

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

#### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

#### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

#### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

#### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

#### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

#### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

#### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

#### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

#### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

#### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

#### Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

#### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

#### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

#### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Combo Box

This object comes from the "Dialog Box" category and can be configured by inserting a list of strings separated by the "|" character, or by setting a query to retrieve data from a DB table. Data is not entered but selected from the drop down menu which opens when clicking the arrow.

## Tag

#### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

#### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Combo Box

### Source Reader Settings

This editor allows you to populate the Combo Box with the items from a DB column.

### Item List

This allows you to add combo box items using a string. Each item must be separated by the 'Pipe' character e.g. 1|2|3

### Tag Item Source

This is used to select the variable from which to obtain the displayed source data

### Text

This is used to set a predefined text to show within the text box.

### Selected Index

This is used to set the index value for the first item on the list.

### Toggle Button Size

This is used to set the size of Toggle Button to open list.

### Read Only

When enabled, this property forces the Combo Box as a read only object and therefore the user will not be able to change the value within it.

### Preserve User Input

When enabled, allows the value entered by the user to be used in cases of matching items on a drop down list.

### Allow Editing

When enabling this property, it will be possible to edit the Combo Box value manually at runtime.

### Keep Open on Edit

Enable this so that the drop-down window stays open while editing the value within the text box.

### Keep Items Synchronized

This allows the various item to stay synchronized with their respective sources.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more languages defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight



In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### Preserve Text

This is used to preserve the added text string when the object is updated from the library.

## User Access

### Access Level

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### User Access Role

This is used to set the name of the user password group to allow user interaction with this object.

### Access Level from Tag

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Group Box

This object is used to frame groups of object items of the same kind. The object Items can be collocated within the frame and the heading (Groupbox) can be changed accordingly.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NEXT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

#### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

#### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

#### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

#### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

#### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

#### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

#### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

#### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

#### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

#### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## User Access

### Access Level

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### User Access Role

This is used to set the name of the user password group to allow user interaction with this object.

### Access Level from Tag

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.



## DB List Box

The Movicon NExT List Box objects are components that can be inserted anywhere on screen in order to display lists of items. A numeric or string type variable can be associated to this object. Each row of the list corresponds to a numeric value that is made available in the variable associated to the object if this variable is numeric type. If the variable is string type, it will be updated with the selected Item's text. Each time a row is selected, the List Box's variable is updated with the corresponding numeric value (the value is progressive, starting from 0 to indicate the first row) or with the selected Item's text.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## ComboBox

### Source Reader Setting

This editor allows you to populate the Combo Box with the items from a DB column.

### Item List

This is used to add combo box items using a string. Each item must be separated by the "Pipe" character, eg. "1|2|3"

### Selected Index

This is used to set the index value for the first listed item.

### Selection Mode

This is used to set the selection mode type of items on the list.

### Keep Items Synchronized

This keeps the selected item synchronized with the relative source.

# Style

## Background Color

This is used to set the container's background color.

## Text Color

This is used to set the object's text color.

## Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

## Vertical Content Alignment

This is used to vertically align all the contents within the object.

## Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

# Fonts

## Font Size

This is used to set the size of the selected font.

## Font Family

This is used to set a font type to use for text.

## Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

## Font Weight

This is used to select the thickness of the selected font.

## Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style

- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style

- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

#### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

#### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

#### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

#### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

#### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## **Animations**

#### **Animations**

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## **Commands**

#### **Commands**

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Option Button

This is a typical Option Button to which a Movicon NExT variable can be associated to obtain a progressive numeric value based on the box selected. For example, the variable will obtain the "0" value when the first box is selected and the "1" when the second box is select and so forth. The number of options to display is set using the object's properties.

The options are selected with a mouse click or by using the keyboard.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Combo Box

### Source Reader Setting

This editor allows you to populate the Combo Box with the items from a DB column.

### Item List

This is used to insert combobox items using a static string.

### Selected Index

This is used to set the index value for the first listed item.

### Bullet Size

This is used to set the size of the selectable box.

### Option Item Size

This is used to set the size of the listed option items.

### Keep Items Synchronized

Enabling this will keep the selected items synchronized with the relative source.



## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size

- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style

- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

#### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

#### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

#### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

#### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

#### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## **Animations**

#### **Animations**

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## **Commands**

#### **Commands**

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Spin Button

This button is used for increasing/decreasing the tag associated to it by one value, specified by the user, using the classic arrow spin function.

This object is usually displayed in other display objects by means of enabling the Enable Spin option.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object. Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Indicator - Basic Aspect

### Engineering Unit

This is used to set the name of the engineering unit. For further information please see chapter: "Using Engineering Units"

### Use Engineering Unit Value

Enable this to use the engineering unit value for the object's Min. and Max. values.

## Indicator – Scale

### Minimum Value

This is used to set the allowed minimum value.

### Maximum Value

This is used to set the allowed maximum value.

### Tag Min. Value

This replaces the minimum scale value with the linked variable value.

### Tag Max Value

This replace the maximum scale value with the linked variable value.

### Value Entry Limits for Controls

The EditDisplay, Gauge, LinearMeter, Slider and Potentiometer control types allow tag values to be displayed and changed. When entering values, you will be able to define the Static Limits (defined in Design mode) or Dynamic Limits (linking two runtime editable tags). In addition, these limits can also be defined in the same Tag by means of the link to an Engineering Unit Tag. The priorities with which these limits are used to intervene depend on certain settings. Below are listed the cases in which these limits may intervene:

5. The Tag is not linked to an Engineering Unit. In this situation, only the Static Limits or the Dynamic Limits are considered. The Dynamic Limits always have priority over Static Limits, therefore, if Tags have been insert to manage the Dynamic Limits in the object, these will be used instead of the Static Limits at Runtime.
6. The Tag is linked to an Engineering Unit and has been set with Static Limits and not Dynamic Limits. In this case, the control's "Enable Engineering Unit" property comes into play. When the property is enabled, the control will inherit the Tag's Engineering Unit's limits and will only use those. Conversely, if the control's "Enable Engineering Unit" property is disabled, the most restrictive limits (both the minimum and maximum) will be considered between the Engineering Units and the Static Limits. For instance, supposing that the Engineering Unit limits are min=0 and max=100 and that the control's Static Limits are min=10 and max=200, the control will then use min=10 and max=100 as limits (the maximum of the minimums and the minimum of the maximums).
7. The Tag has been linked with a Engineering Unit and set with Dynamic Limits. In this case, the most restrictive limits will always be considered (both minimum and maximum) between the Engineering Unit and the Dynamic Limits. For instance, supposing the Engineering Units are min=10 and max=100 and that the control's Dynamic Limits are min=5 and max=90, the control will then use
8. min=10 and max=90 as limits (the maximum of the minimums and the minimum of the maximums).

## Spin Control

### Spin Step

This is used to set the increase/decrease imposta il passo per l'incremento/decremento valore.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment



This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### Force as Dynamic on Client

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

#### Access Level from Tag

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

#### Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

#### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

#### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

#### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## 2.12. Displays

The Movicon NExT Displays are components which can be inserted anywhere on screen for dynamically displaying variables.

The display object is used for showing figures or strings whose value is the content of the associated variable that can be set and configured using the 'properties window'.

The Display objects are available from the ToolBox "Displays" Group.

The Display object also allows the operator to modify the value of the associated variable. This is done by selecting the Display and entering the value desired or using the spin arrow command at the side to increase or decrease the value being displayed. The Display object can also be set to read only using the appropriate property.



The CalendarEditDisplay object is used to select a date but not the time. If a format is defined in the "Display Format String" property that includes the time as well, this format will be applied so that when selecting a date, the hour will also be added and declared as 00.00:00.

The Display shows the value of the variable assigned during project runtime time with the style and size set using the its associated properties.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

### Stat Def

When a variable has been enabled with statistics and inserted in the Tag field, it is possible to select one of below Tag values to display in the Display:

- None: current variable value (default)
- Min: minimum value assumed by the variable.
- Max: maximum value assumed by the variable.
- Average: variable's calculated average value.
- NumUpdate: number of variations undergone by the variable.
- TotalTimeOn: the total time with which the variable maintained a value other than zero.



Use the "Time Span Format" property to customize how the value is to be displayed.

## Limits

### Minimum Value

This is used to set the allowed minimum value.

### Maximum Value

This is used to set the allowed maximum value.

### Tag Min. Value

This replaces the minimum scale value with the linked variable value.

### Tag Max Value

This replace the maximum scale value with the linked variable value.

### Value Entry Limits for Controls

The EditDisplay, Gauge, LinearMeter, Slider and Potentiometer control types allow tag values to be displayed and changed. When entering values, you will be able to define the Static Limits (defined in Design mode) or Dynamic Limits (linking two runtime editable tags). In addition, these limits can also be defined in the same Tag by means of the link to an Engineering Unit Tag. The priorities with which these limits are used to intervene depend on certain settings. Below are listed the cases in which these limits may intervene:

1. The Tag is not linked to an Engineering Unit. In this situation, only the Static Limits or the Dynamic Limits are considered. The Dynamic Limits always have priority over Static Limits, therefore, if Tags have been insert to manage the Dynamic Limits in the object, these will be used instead of the Static Limits at Runtime.
2. The Tag is linked to an Engineering Unit and has been set with Static Limits and not Dynamic Limits. In this case, the control's "Enable Engineering Unit" property comes into play. When the property is enabled, the control will inherit the Tag's Engineering Unit's limits and will only use those. Conversely, if the control's "Enable Engineering Unit" property is disabled, the most restrictive limits (both the minimum and maximum) will be considered between the Engineering Units and the Static Limits. For instance, supposing that the Engineering Unit limits are min=0 and max=100 and that the control's Static Limits are min=10 and max=200, the control will then use min=10 and max=100 as limits (the maximum of the minimums and the minimum of the maximums).
3. The Tag has been linked with a Engineering Unit and set with Dynamic Limits. In this case, the most restrictive limits will always be considered (both minimum and maximum) between the Engineering Unit and the Dynamic Limits. For instance, supposing the Engineering Units are min=10 and max=100 and that the control's Dynamic Limits are min=5 and max=90, the control will then use
4. min=10 and max=90 as limits (the maximum of the minimums and the minimum of the maximums).

## Calendar Display

### Null Text

This is used to set a text to display when the value is "null".

### Read Only

This is used to set the object as 'Read Only'.

### Display Format String

This is used to set the text string Format to display (calendar).

## Date-Time Display

### Read Only

This is used to set the object as 'Read Only'.

### Show Border

Shows or hide the object's border.



### **Null Text**

This is used to set a text to display when the value is "null".

### **Display Format String**

This is used to set the text format to display. For further information please also see Standard format strings and Custom format strings

### **Convert Format String**

This is used to convert the format.

## **Display Settings**

### **Spin Enabled**

This is used to activate/deactivate the spin function.

### **Read Only**

This is used to set the object as 'Ready Only'.

### **Unit Label**

This is used to set a unit of measure label in the Display. When this is not used, the measure unit defined in the Engineering Unit associated to the variable will be used if defined.

### **Show Thousands Separator**

This property is used to display a thousands separator in the display.

### **Border Thickness**

This is used to set the display's border thickness in pixels.

### **Spin Position**

This is used to move the position of the spin in the display.

### **Spin Size**

This is used to change the size of the spin.

### **Spin Step**

This is used to set the increase/decrease step value for the spin button.

### **Enable Out of Range Icon**

This is used to enable the out of value range management.

### **Use Engineering Unit Limits**

Enabling this object property, the object will use the limits defined by the Engineering Unit linked to the Tag and will ignore the set the static limits it was set with. Otherwise, the limits used will be the most restrictive minimum and maximum limits between the Static Limits and that of the Engineering Unit linked to the Tag.

This property will be ignored when the object has been set with Dynamic Limits.

In addition, when this property is enabled, the object will also inherit the Unit of Measure defined in the Engineering Unit linked to the Tag. When the Unit of Measure is also defined in the object, this will have priority over the one defined in the Engineering Unit.

### Password Style Digits

Enabling this will allow you hide the value in "Password" style.

### Precision Digits

This is used to set the number of decimal digits to display.



When floating point variable types are associated to the display, this parameter is used to determine how many decimal digits are to be displayed. When integer variable types are associated to the display, this parameter is used to display the value with a floating point. For example, when setting this field to 2 and the associated integer variable value is 123, it will be shown in the Display object as '1.23'. In this case, the minimum and maximum limits and the spin step will not take into account the format set in the display but, instead, will always work with the variable's integer value. For example, with a decimal digit and a maximum limit of '100', it will be possible to edit the variable with a maximum value equal to '100' which will therefore show in the Display object as '10.0'.

### Time Span Format

This is used to specify a custom format to display the value for the "Total Time On" statistics.

- **Standard format strings for Date/Time values**

The standard Date/Time format strings contain one character. This character defines the model used for representing the value (and in which mode to display the number of years, months and so on).

Format Specifier	Description	Result of formatting a sample value (March 25 2019 12:00:00)
d	Short date pattern	3/25/2019
D	Long date pattern	Monday, March 25, 2019
t	Short time pattern	12:00 AM
T	Long time pattern	12:00:00 AM
f	Full date/time pattern	Monday, March 25, 2003 12:00 AM
F	Full date/time pattern	Monday, March 25, 2003 12:00:00 AM
g	General date/time pattern	3/25/2019 12:00 AM

G	General date/time pattern	3/25/2019 12:00:00 AM
---	---------------------------	-----------------------

- **Custom format Strings for Date/Time**

To create models for Date/Time values, you will need to combine the strings listed in the table below. These strings represent the year, month, day, number and so on in different formats.

<b>Symbol</b>	<b>Meaning</b>	<b>Result of formatting a sample value (March 25 2019 12:00:00)</b>
yy	The last two digits of the year.	19
yyyy	A four digit year	2019
MM	The number of the month	03
MMM	A short text description of the month	Mar
MMMM	The full name of the month	March
dd	The number of the day	25
ddd	A short text for the day of the week	Mon
dddd	The full name of the day of the week	Monday
/	Date separator. Its appearance depends on regional settings.	

<b>Symbol</b>	<b>Meaning</b>
hh	Hours
mm	Minutes
ss	Seconds
tt	If present represent data in AM/PM format

:	Time separator
---	----------------

- **Custom format Strings for TotalTimeON statistic values**

Symbol	Meaning
hh	Hours
mm	Minutes
ss	Seconds
f	Tenths of Seconds
ff	Hundredths of a second
fff	Milliseconds



You must always use "\" separator character.

Example:

- dd\hh:mm -> 30.10:20
- hh:mm:ss -> 10:20:52
- mm:ss -> 20:52
- dd\hh:mm:ss\fff -> 30.10:20:52.657



Customized DateTime format settings might not be supported in WebHMI.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

### Out of Range Icon Color

This is used to set the color of the out of range icon when displayed.

### Border Color

This is used to select the color of the display's border color.

### Spin Color

This is used to change the color of the spin button.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more languages defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed.

#### **Horizontal Text Align.**

This is used to change the text's vertical alignment.

#### **Vertical Text Align.**

This is used to change the text's vertical alignment.

#### **Unit Offset**

Ready only value.

#### **Text Offset**

This is used to set the Imposta il Valore dello Spin per il quale si andrà a modificare il valore visualizzato.

#### **Text Wrapping**

This is used to activate/deactivate the text wrapping to bump down text to the next line when overflowing the display object.

## **Layout**

#### **Position X**

This is used to set the object's horizontal position in pixels.

#### **Position Y**

This is used to set the object's vertical position in pixels.

#### **Height**

This is used to change the object's height.

#### **Width**

This is used to change the object's width.

#### **Border Thickness**

This is used to set the container's border thickness.

#### **Margin**

This is used to set the object's margins.

#### **Lock Object Position**

This locks the object's position on screen.

## **Visibility**

#### **Transparency**

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### Force as Dynamic on Client

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight



In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### Preserve Text

This is used to preserve the added text string when the object is updated from the library.

## User Access

### Access Level

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### User Access Role

This is used to set the name of the user password group to allow user interaction with this object.

### Access Level from Tag

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## 2.13. Advanced Gauges and Arc Bargraphs

### Gauges (Evolved)

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

A Gauge comes with a Scale, Bar and indicator that can all be configured using the relative associated properties.

### Arc Bargraph

The Movicon NExT dynamic arcs are configurable objects that display variable values graphically.

A dynamic Arc consists of a Scale and a Bar which can be configured using the relevant properties.



The graphical appearance of these objects can be customized using the theme applications available in the relative style properties of the various components.

### Tag

#### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

#### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

### Indicator - Basic Aspect

#### Gauge Style

This is used to set the basic style of the Gauge/Bargraph

#### Gauge Back Color

This is used to set the back color for the object's back color.

#### Scale Line Color

This is used to set the color of the scale line.

#### Start Angle

This is used to set where the angle starts.

### End Angle

This is used to set where the angle ends.

### Engineering Unit

This is used to set the name of the engineering unit. For further information, please see: "Using Engineering Units"

### Show Value

Enabling this will show the exact value indicated by the indicator at that precise moment.

### Use Engineering Unit Value

This allows the engineering unit values to be used for the object's min. and max. value.

### Show Engineering Unit

This is used to enable/disable the engineering unit label.

### Value Offset

This is used to set the value offset.

### Value Format

This is used to set which value the format is to be displayed with.

### Auto Hide Elements

When this property is set to true, the elements will gradually disappear and the number of ticks will lessen as the control is made smaller and as a consequence so will the labels. For example, as the object is gradually made smaller, the numerical values will start to disappear in the order they are shown, along with the ticks, the scale numeration, the indicator and the border and so forth.

When the property is set to False, all the elements will remain visible when the object is reduced in size but may overlap each other as a consequence.



This behaviour is determined by the fact that when re-sizing the object, the elements, such as the scale and fonts, do not re-size accordingly but remain their initial set sizes.

If you wish to resize the object so that all its elements re-size accordingly, you will need create a symbol with the object's initial sizes, and then when you re-size it (so that it becomes a ViewBox) all its elements will re-size proportionately.

### Use Engineering Unit Limits

When enabling this property, the object will use the limits defined in the Engineering Unit linked to the Tag and will ignore the static limits it was set with. Otherwise, the limits used will be those given by the most restrictive minimum and maximum limits between the Static Limits and the Engineering Unit associated to the Tag.

This property will be ignored when the object has been set with Dynamic Limits.

In addition, when this property is enabled, the object will also inherit the Unit of Measure set in the Engineering Unit linked to the Tag. When the Unit of Measure has also been set in that object, this will have priority over the one set in the Engineering Unit.

For further information, please see "Using Engineering Units"

## Indicator - Scale

### Minimum Value

This is used to set the allowed minimum value.

### Maximum Value

This is used to set the allowed maximum value.

### Tag Min. Value

This replaces the minimum scale value with the linked variable value.

### Tag Max Value

This replace the maximum scale value with the linked variable value.

### Value Entry Limits for Controls

The EditDisplay, Gauge, LinearMeter, Slider and Potentiometer control types allow tag values to be displayed and changed. When entering values, you will be able to define the Static Limits (defined in Design mode) or Dynamic Limits (linking two runtime editable tags). In addition, these limits can also be defined in the same Tag by means of the link to an Engineering Unit Tag. The priorities with which these limits are used to intervene depend on certain settings. Below are listed the cases in which these limits may intervene:

1. The Tag is not linked to an Engineering Unit. In this situation, only the Static Limits or the Dynamic Limits are considered. The Dynamic Limits always have priority over Static Limits, therefore, if Tags have been insert to manage the Dynamic Limits in the object, these will be used instead of the Static Limits at Runtime.
2. The Tag is linked to an Engineering Unit and has been set with Static Limits and not Dynamic Limits. In this case, the control's "Enable Engineering Unit" property comes into play. When the property is enabled, the control will inherit the Tag's Engineering Unit's limits and will only use those. Conversely, if the control's "Enable Engineering Unit" property is disabled, the most restrictive limits (both the minimum and maximum) will be considered between the Engineering Units and the Static Limits. For instance, supposing that the Engineering Unit limits are min=0 and max=100 and that the control's Static Limits are min=10 and max=200, the control will then use min=10 and max=100 as limits (the maximum of the minimums and the minimum of the maximums).
3. The Tag has been linked with a Engineering Unit and set with Dynamic Limits. In this case, the most restrictive limits will always be considered (both minimum and maximum) between the Engineering Unit and the Dynamic Limits. For instance, supposing the Engineering Units are min=10 and max=100 and that the control's Dynamic Limits are min=5 and max=90, the control will then use
4. min=10 and max=90 as limits (the maximum of the minimums and the minimum of the maximums).

## Indicator - Marker

### Show Marker

This is used to display/hide the marker.

### **Marker Color**

This is used to set the color of the marker.

### **Marker Border Color**

This is used to set the marker's border color.

### **Marker Height**

This is used to set the height of the marker.

### **Marker Width**

This is used to set the width of the marker.

### **Marker Offset**

This is used to set the space between the marker and the border.

### **Marker Animation**

This is used to activate/deactivate the marker's animation.

### **Marker Clickable**

This is used to activate/deactivate the marker's interactivity. When activated the user will be able to set the value directly by clicking and moving the marker.

## **Indicator - Dividers**

### **Tickmark Style**

This is used to set the object's tickmark style.

### **Major Tickmark Color**

This is used to set the color of the major tickmarks.

### **Minor Interval Count**

This is used to set the number of minor scale tickmarks.

### **Major Tickmark Length**

This is used to set the major tickmarks on the object's scale.

### **Minor Tickmark Color**

This is used to set the color of the minor tickmarks.

### **Major Interval Count**

This is used to set the number of major scale tickmarks.

### **Minor Tickmark Length**

This is used to set the length of the minor tickmarks.

#### **Major Tickmark Thickness**

This is used to set the thickness of the major tickmarks.

#### **Major Tickmark Offset**

This is used to set the distance of the major tickmarks from the object's border.

#### **Show First Major Tickmark**

Shows/Hides the scale's first major tickmark.

#### **Show Last Major Tickmark**

Shows/Hides the scale's last major tickmark.

#### **Minor Tickmark Offset**

This is used to set the distance of the minor tickmarks from the object's border.

#### **Minor Tickmark Thickness**

This is used to set the thickness of the minor tickmarks.

#### **Show Minor Tickmarks on Major**

This enables the minor tickmarks to show in the same position of a major tickmark.

## **Indicator - Range Bar**

#### **Show Range Bar**

Enables/disables the Value Range Bar.

#### **Range Bar Color**

This is used to set the color of the Value Range Bar.

#### **Range Bar Background**

This is used to set the background color of the Range bar.

#### **Range Bar Thickness**

This is used to set the Range Bar's thickness.

#### **Range Bar Offset**

This is used to set the space between the Range Bar segments.

#### **Range Bar Animation**

Enables/disables the Range Bar animation. When enabled, the bar will fill until it reaches the relative position during animation.

### Range Bar Clickable

Enables/disables interactivity with the Range Bar.

## Indicator - Labels

### Label Color

This is used to set the color of the label.

### Value Label Color

This is used to set the color of the values on the labels.

### Engineering Unit Label Color

This is used to select the engineering unit label's text color.

### Engineering Unit Label Offset

This is used to set the position of the engineering unit labels.

### Label Orientation

This is used to set the scale's label orientation.



This property is not supported in WebHMI and WebClient (SVG/HTML5)

### Label Offset

This is used to set the distance of the scale's labels from the border.

### Label Text Format

This is used to set the label's text format which for default is {0:0}. If you wish to keep two digits after the point for example, you should set this field as {0:0.00}.

### Show First Label

Enabling this will show the first scale's label.

### Show Last Label

Enabling this will show the last scale's label.

## Indicator - Needle

### Show Needle

This is used to show/hide the needle.

### Needle Style

This is used to set the needle's basic style.



### **Needle Color**

This is used to set the color of the needle.

### **Needle Start Offset**

This is used to set the distance between the needle and the object's center in pixels.

### **Needle End Offset**

This is used to set the distance between the point of the needle (end) and the object's border in pixels.

### **Needle Clickable**

Activates/Deactivates interactivity with needle. When activated, the user will be able to set the value directly by clicking and moving the needle.

### **Needle Animation**

Activates/deactivated the needle's animation.

### **Needle Layout Index**

This is used to set the needle's depth level.

## **Indicator - Alarm Zone**

### **Alarm Zone Visible**

This is used to make the preselected scale zone (zones 1,2 or 3) visible/invisible.

### **Alarm Zone Start**

This is used to set the start position of the preselected scale zone (zones 1,2 or 3).

### **Alarm Zone End**

This is used to set the end position of the preselected scale zone (zones 1, 2 or 3).

### **Alarm Zone Thickness**

This is used to set the zone's thickness (zones 1, 2, 3).

### **Alarm Zone Offset**

This is used to set the space between the scale zones (zones 1, 2, 3).

### **Alarm Zone Color**

This is used to set the color of the alarm zones (zones 1, 2, 3).

## **Indicator - Pivot**

### **Pivot Style**

This is used to set the style of the pivot in the centre of the object.

#### **Pivot Color**

This is used to select a color for the pivot.

#### **Pivot Height**

This is used to select the height of the pivot in the centre.

#### **Pivot Width**

This is used to set the width of the pivot in the centre.

## **Spin Control**

#### **Minimum Value**

This is used to set the minimum value allowed.

#### **Maximum Value**

This is used to set the maximum value allowed.

## **Style**

#### **Background Color**

This is used to set the container's background color.

#### **Text Color**

This is used to set the object's text color.

#### **Horizontal Content Alignment**

This is used to horizontally align all the contents within the object.

#### **Vertical Content Alignment**

This is used to vertically align all the contents within the object.

#### **Disable Anti-Aliasing**

This is used to Enable/Disable the Anti-aliasing function.

## **Fonts**

#### **Font Size**

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more languages defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

### Label Font

This is used to set the font of the labels on the scale.

### Value Label Font

This is used to set the font of the values.

### Engineering Unit Font

This is used to set the font of the engineering unit labels.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### Force as Dynamic on Client

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

#### Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

#### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

#### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

#### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## 2.14. Gauges

Movicon NExT's Fast Gauges are configurable objects that allow the defined variable values to be displayed graphically.

A Fast Gauge comes with a scale, bar and needle that can be configured using the relevant associated properties.

As a difference to the classic Gauge, this version is 'lighter' to load on screen and more performing even though limited in its range of configurations.

## Tag



### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Indicator - Basic Aspect

### Gauge Style

This is used to set the Gauge/Bargraph's base style.

### Gauge Scale Radius

This allows you to set the percentage of gauge scale's radius.

### Initial Angle

This is used to set the scale's initial angle measure.

### Final Angle

This is used to set the scales final angle measure.

### Engineering Unit

This is used to set the name of the engineering unit. For further information, please refer to the topic on "Using Engineering Units"

### Show Engineering Unit

Use this to enable/disable the engineering unit label.

### Show Value

Enabling this will show the exact value being indicated by the needle in the object at that current time.

### Use Engineering Unit Limits

When enabling this property, the object will use the limits defined in the Engineering Unit linked to the Tag and will ignore the static limits it was set with. Otherwise, the limits used will be those given by the most restrictive minimum and maximum limits between the Static Limits and the Engineering Unit associated to the Tag.

This property will be ignored when the object has been set with Dynamic Limits.

In addition, when this property is enabled, the object will also inherit the Unit of Measure set in the Engineering Unit linked to the Tag. When the Unit of Measure has also been set in that object, this will have priority over the one set in the Engineering Unit.

For further information, please see "Using Engineering Units"

### Use Engineering Unit Limits

This allows the engineering unit's values to be used as the object's Min. and Max. values.

### Value Text Spacing (Advanced)

This is used to set the space between the labels border and the text in pixels.

### Numeric Value Format (Advanced)

This is used to set the value display format.

## Indicator - Scale

### Minimum Value

This is used to set the allowed minimum value.

### Maximum Value

This is used to set the allowed maximum value.

### Tag Min. Value

This replaces the minimum scale value with the linked variable value.

### Tag Max Value

This replace the maximum scale value with the linked variable value.

### Value Entry Limits for Controls

The EditDisplay, Gauge, LinearMeter, Slider and Potentiometer control types allow tag values to be displayed and changed. When entering values, you will be able to define the Static Limits (defined in Design mode) or Dynamic Limits (linking two runtime editable tags). In addition, these limits can also be defined in the same Tag by means of the link to an Engineering Unit Tag. The priorities with which these limits are used to intervene depend on certain settings. Below are listed the cases in which these limits may intervene:

1. The Tag is not linked to an Engineering Unit. In this situation, only the Static Limits or the Dynamic Limits are considered. The Dynamic Limits always have priority over Static Limits, therefore, if Tags have been insert to manage the Dynamic Limits in the object, these will be used instead of the Static Limits at Runtime.
2. The Tag is linked to an Engineering Unit and has been set with Static Limits and not Dynamic Limits. In this case, the control's "Enable Engineering Unit" property comes into play. When the property is enabled, the control will inherit the Tag's Engineering Unit's limits and will only use those. Conversely, if the control's "Enable Engineering Unit" property is disabled, the most restrictive limits (both the minimum and maximum) will be considered between the Engineering Units and the Static Limits. For instance, supposing that the Engineering Unit limits are min=0 and max=100 and that the control's Static Limits are min=10 and max=200, the control will then use min=10 and max=100 as limits (the maximum of the minimums and the minimum of the maximums).
3. The Tag has been linked with a Engineering Unit and set with Dynamic Limits. In this case, the most restrictive limits will always be considered (both minimum and maximum) between the Engineering Unit and the Dynamic Limits. For instance, supposing the Engineering Units are min=10 and max=100 and that the control's Dynamic Limits are min=5 and max=90, the control will then use

4. min=10 and max=90 as limits (the maximum of the minimums and the minimum of the maximums).

## Indicator - Marker

### Show Marker

Shows/Hides the Visualizza/Nasconde la punta della lancetta.

### Marker Color

This is used to set the color of the marker.

### Marker Border Color

This is used to set the color of the marker's border.

### Marker height

This is used to set the marker's height.

### Marker Width

This is used to set the marker's width.

### Marker Style

This is used set the marker's style.

### Marker border thickness (Advanced)

This is used to set the marker border's thickness.

### Marker Spacing (Advanced)

This is used to set the space between the marker and the border.

### Marker Position (Advanced)

This is used to set where to place the marker.

## Indicator - Tickmarks

### Tickmark Position

This is used to set the where to place the tickmarks.

### Tickmark Color

This is used to set the tickmark color.

### Number of Major Tickmarks

This is used to set the number of major tickmarks to show on the scale.

### Number of Minor Tickmarks

This is used to set the number of mine tickmarks to show on the scale.

### Major Tickmark Length

This is used to set the length of the object's major scale tickmarks.

### Minor Tickmark Length

This is used to set the length of the object's minor scale tickmarks.

### Major Tickmark Spacing (Advanced)

This is used to set the space between the object's major scale tickmarks and its border.

### Minor Tickmark Spacing (Advance)

This is used to set the space between the object's minor scaled tickmarks and its border.

### Major Tickmark Thickness (Advanced)

This is used to set the thickness of the major tickmarks.

### Minor Tickmark Thickness (Advanced)

This is used to set the thickness of the minor tickmarks.

## Indicator - Range Bar

### Range Bar Color

This is used to set the color of the Range Bar.

### Range Bar Background Color

This is used to set the background color of the Range Bar.

### Range Bar Thickness

This is used to set the thickness of the Range Bar.

## Indicator - Labels

### Label Color

This is used to set the color of the scale labels.

### Value Label Color

This is used to set the color of the values on the labels.

### EU Label Color

This is used to select a test color for the engineering unit label.

#### **Label Spacing (Advanced)**

This is used to set the space between the scale labels and the border.

#### **EU Label Offset (Advanced)**

This is used to set the engineering unit labels' offset from the scale tickmarks.

#### **Label Text Format (Advanced)**

This is used to set the label text format, which is set to (0:0) for default. If you wish to keep the two figures after the decimal point, use this setting:{0:0.00}

## **Indicator - Needle**

#### **Show Needle**

This is used to hide/show the needle.

#### **Needle Style**

This is used to set the needle's style.

#### **Needle Color**

This is used to set the needle's color.

#### **Needle Height**

This is used to set the needle's height.

#### **Needle Width**

This is used to set the needle's width.

## **Indicator - Spindle Cap**

#### **Show Spindle Cap**

Ticking this option box will show the spindle cap.

#### **Spindle Cap Style**

This is used to set the spindle cap's style.

#### **Spindle Cap Color**

This is used to set the spindle cap's color.

#### **Spindle Cap Radius**

This is used to set the spindle cap's radius.

# Style

## Background Color

This is used to set the container's background color.

## Text Color

This is used to set the object's text color.

## Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

## Vertical Content Alignment

This is used to vertically align all the contents within the object.

## Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

# Fonts

## Font Size

This is used to set the size of the selected font.

## Font Family

This is used to set a font type to use for text.

## Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

## Font Weight

This is used to select the thickness of the selected font.

## Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style

- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

### Font Etichetta Valore

Permette di impostare lo stile del carattere del valore attuale.

### Font Unità Ingegneristica

Permette di impostare lo stile del carattere dell'etichetta riguardante l'unità ingegneristica.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.



### **Text Direction**

This allows you to set the direction of the text within the object.

### **Enable UI**

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

#### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

#### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

#### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

#### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

#### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## **Animations**

#### **Animations**

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## **Commands**

#### **Commands**

This property gives access to the window containing commands that can be associated to this object. These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## 2.15. Fast Meters

The Movicon NExT "Fast Meters" are configurable objects that allow specified variable values to be displayed graphically. "Fast Meter" objects are build with a "Scale" and "indicator" that can be configured by setting the appropriate properties as desired.

Compared to the classic "Meter" objects, these "Fast Meter" objects are lighter to load on screen thus more performing despite limited in its range of configurations.

### Tag

#### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

#### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

### Indicator - Basic Aspect

#### Meter Style

This is used to set the Scale's base style.

#### Meter Orientation

This is used to set the object's orientation.

#### Engineering Unit

This is used to set the name of the engineering unit. For further information, please refer to "Using Engineering Units".

#### Show Engineering Unit

Use this to enable/disable the engineering unit label.

#### Show Value

Shows the label with the current value associated to the Tag.

### Use Engineering Unit Limits

When enabling this property, the object will use the limits defined in the Engineering Unit linked to the Tag and will ignore the static limits it was set with. Otherwise, the limits used will be those given by the most restrictive minimum and maximum limits between the Static Limits and the Engineering Unit associated to the Tag.

This property will be ignored when the object has been set with Dynamic Limits.

In addition, when this property is enabled, the object will also inherit the Unit of Measure set in the Engineering Unit linked to the Tag. When the Unit of Measure has also been set in that object, this will have priority over the one set in the Engineering Unit.

For further information, please see "Using Engineering Units" .

### Value Text Offset (Advanced)

This allows you to set the space between the text and the label border.

### Numeric Value Format (Advanced)

This is used to set the number format (e.g. 000.0).

## Indicator - Scale

### Minimum Value

This is used to set the allowed minimum value.

### Maximum Value

This is used to set the allowed maximum value.

### Tag Min. Value

This replaces the minimum scale value with the linked variable value.

### Tag Max Value

This replace the maximum scale value with the linked variable value.

### Value Entry Limits for Controls

The EditDisplay, Gauge, LinearMeter, Slider and Potentiometer control types allow tag values to be displayed and changed. When entering values, you will be able to define the Static Limits (defined in Design mode) or Dynamic Limits (linking two runtime editable tags). In addition, these limits can also be defined in the same Tag by means of the link to an Engineering Unit Tag. The priorities with which these limits are used to intervene depend on certain settings. Below are listed the cases in which these limits may intervene:

1. The Tag is not linked to an Engineering Unit. In this situation, only the Static Limits or the Dynamic Limits are considered. The Dynamic Limits always have priority over Static Limits, therefore, if Tags have been insert to manage the Dynamic Limits in the object, these will be used instead of the Static Limits at Runtime.
2. The Tag is linked to an Engineering Unit and has been set with Static Limits and not Dynamic Limits. In this case, the control's "Enable Engineering Unit" property comes into play. When the property is enabled, the control will inherit the Tag's Engineering Unit's limits and will only use those. Conversely, if the control's "Enable Engineering Unit" property is disabled, the most

restrictive limits (both the minimum and maximum) will be considered between the Engineering Units and the Static Limits. For instance, supposing that the Engineering Unit limits are min=0 and max=100 and that the control's Static Limits are min=10 and max=200, the control will then use min=10 and max=100 as limits (the maximum of the minimums and the minimum of the maximums).

3. The Tag has been linked with a Engineering Unit and set with Dynamic Limits. In this case, the most restrictive limits will always be considered (both minimum and maximum) between the Engineering Unit and the Dynamic Limits. For instance, supposing the Engineering Units are min=10 and max=100 and that the control's Dynamic Limits are min=5 and max=90, the control will then use
4. min=10 and max=90 as limits (the maximum of the minimums and the minimum of the maximums).

## Indicator - Marker

### Show Marker

This option allows you to display or hide the marker.

### Marker Color

This is used to set the color of the marker.

### Marker Border Color

This is used to set the color of the marker's border.

### Marker height

This is used to set the marker's height.

### Marker Width

This is used to set the marker's width.

### Marker Style

This is used set the marker's style.

### Marker Spacing (Advanced)

This is used to set the space between the marker and the border.

### Marker border thickness (Advanced)

This is used to set the marker border's thickness.

## Indicator - Tickmarks

### Tickmark Color

This is used to set the tickmark color.

### Number of Major Tickmarks

This is used to set the number of major tickmarks to show on the scale.

### Number of Minor Tickmarks

This is used to set the number of mine tickmarks to show on the scale.

### Major Tickmark Length

This is used to set the length of the object's major scale tickmarks.

### Minor Tickmark Length

This is used to set the length of the object's minor scale tickmarks.

### Major Tickmark Spacing (Advanced)

This is used to set the space between the object's major scale tickmarks and its border.

### Minor Tickmark Spacing (Advance)

This is used to set the space between the object's minor scaled tickmarks and its border.

### Major Tickmark Thickness (Advanced)

This is used to set the thickness of the major tickmarks.

### Minor Tickmark Thickness (Advanced)

This is used to set the thickness of the minor tickmarks.

## Indicator - Range Bar

### Range Bar Color

This is used to set the color of the Range Bar.

### Range Bar Background Color

This is used to set the background color of the Range Bar.

### Range Bar Thickness

This is used to set the thickness of the Range Bar.

## Indicatore - Labels

### Label Color

This is used to set the color of the scale labels.

### Value Label Color

This is used to set the color of the values on the labels.

### EU Label Color

This is used to select a test color for the engineering unit label.

#### **Label Spacing (Advanced)**

This is used to set the space between the scale labels and the border.

#### **EU Label Offset (Advanced)**

This is used to set the engineering unit labels' offset from the scale tickmarks.

#### **Label Text Format (Advanced)**

This is used to set the label text format, which is set to (0:0) for default. If you wish to keep the two figures after the decimal point, use this setting:{0:0.00}

## **Style**

#### **Background Color**

This is used to set the container's background color.

#### **Text Color**

This is used to set the object's text color.

#### **Horizontal Content Alignment**

This is used to horizontally align all the contents within the object.

#### **Vertical Content Alignment**

This is used to vertically align all the contents within the object.

#### **Disable Anti-Aliasing**

This is used to Enable/Disable the Anti-aliasing function.

## **2.16. Hot Region**

The Hot Regions constitute zones that can be strategically inserted and positioned on screen and remain invisible but mouse sensitive during project runtime.

Like the most of the objects the Hot Regions have execution properties for setting associated commands when the operator clicks within the Hot Region area.

When a Hot region is inserted on screen it will appear as a shaded area with default sizes. Once inserted it can be sized as pleased by dragging its borders with the mouse.



The zone of the screen within which the Hot Region is placed, is usually built with a background zone representing a strategic drawing. This zone will be invisible during project runtime but the Hot Region borders will highlight when the mouse pointer



hovers within its boundaries. ToolTips can also be made to appear at the same time this happens if desired by the programmer.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Button Settings

### Click Mode

This is used to set which click mode to generate the command with the button.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NEXT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

**Height**

This is used to change the object's height.

**Width**

This is used to change the object's width.

**Border Thickness**

This is used to set the container's border thickness.

**Margin**

This is used to set the object's margins.

**Lock Object Position**

This locks the object's position on screen.

## Visibility

**Transparency**

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

**Visibility**

This is used to enable the object's visibility.

**Visible on Web Client**

This makes the object visible on Web Client.

**Visibility Layer Level**

This is used to set the mask that determines the object's visibility.

**Zoom Level Visibility X**

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y**

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

**ToolTip**

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### Force as Dynamic on Client

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

### Preserve Size

This is used to preserve the values of the local size property group when the object is updated from the library.

### Preserve Commands

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## 2.17. Dynamic Image Object

The "Dynamic Image" object is used to set a box with an animated image. The image in the box set by the user can also be associated to a variable, or eventually, taken directly from the Image List Editor using the **Smart TAG** settings (located at the end of the object's list of properties).

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Dynamic Text

### Animation Time

The is used to set a pause time between one animated image and the next.

### Control Render transform Origin

This is used to define the animation's render transform origin.

### Stretch Type

This is used to set the animated object's content to that it remains within the borders.

### Object Configuration

For further details please see: Smart Properties

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.



### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### **Menu Activation**

This allows you to set the name of the contextual menu that shows when right clicking the object.

### **Is Interactive**

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### **Menu with Left button**

This enables the contextual menu to show by also using the left mouse key.

### **TAB Stop**

This makes the object selectable at runtime using the TAB button.

### **Text Direction**

This allows you to set the direction of the text within the object.

### **Enable UI**

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

### Preserve Size

This is used to preserve the values of the local size property group when the object is updated from the library.

### Preserve Commands

This is used to preserve the list of commands when the object is updated from the library.

### Preserve Animations

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

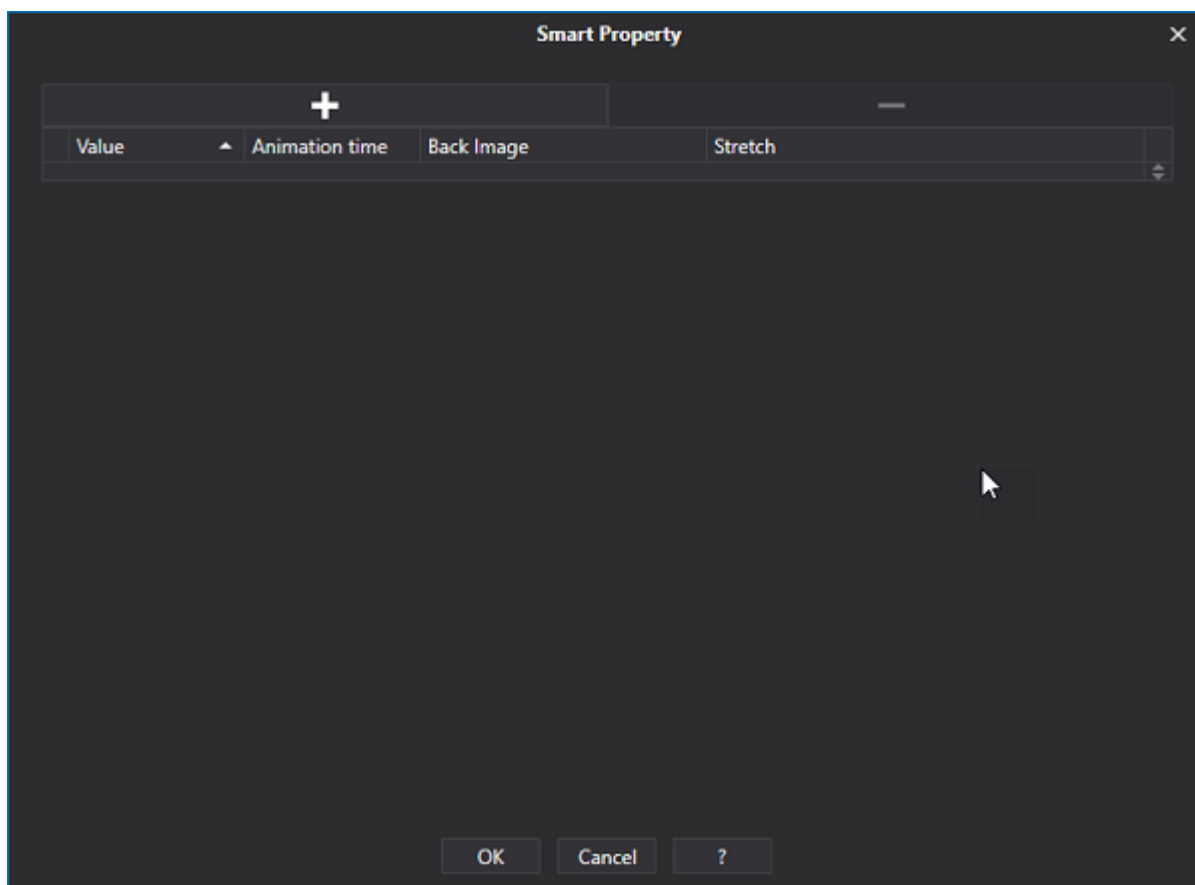
This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Smart Properties



### Value

The threshold of the reached value to activate the animation.

### Animation Time

Scroll time when two or more images have been set in the back image.

### Back Image

This is used to set the object with a back image.

### Stretch

Stretches the image to the size of the container.

## 2.18. Multimedia

### IP Camera Viewer

The IP Camera Viewer allows videos to be streamed at runtime that is transmitted by IP cameras using the MJPEG protocol.

Motion JPEG (M-JPEG) is a video codec in which each frame of the video is compressed to a JPEG image.



There is a Streaming Image Viewer object available to view video streaming with other protocols.



The object can be connected with protected link by setting the required credentials using the appropriate authentication properties.

### Tag

#### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

#### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

### IP Camera Settings

#### IP Camera URL Address

This is used to set the URL address for the IP camera images.

#### Keep Aspect Ratio

Enabling this option will keep the displayed image unchanged.

#### **URL User ID**

This is used to set a URL address User ID when the connection requires a password.

#### **IP Camera User Password**

This is used to set the IP Camera user password to connect user.

#### **Disconnection Timeout**

This is used to set the maximum time to wait in milliseconds before disconnecting.

#### **Reconnection Time**

This is used to set the time to wait in milliseconds before reconnecting.

#### **Image Update Rate**

This is used to set the onscreen image refresh rate in milliseconds.

## **Style**

#### **Background Color**

This is used to set the container's background color.

#### **Text Color**

This is used to set the object's text color.

#### **Horizontal Content Alignment**

This is used to horizontally align all the contents within the object.

#### **Vertical Content Alignment**

This is used to vertically align all the contents within the object.

#### **Disable Anti-Aliasing**

This is used to Enable/Disable the Anti-aliasing function.

## **Fonts**

#### **Font Size**

This is used to set the size of the selected font.

#### **Font Family**

This is used to set a font type to use for text.



### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### **Menu Activation**

This allows you to set the name of the contextual menu that shows when right clicking the object.

### **Is Interactive**

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### **Menu with Left button**

This enables the contextual menu to show by also using the left mouse key.

### **TAB Stop**

This makes the object selectable at runtime using the TAB button.

### **Text Direction**

This allows you to set the direction of the text within the object.

### **Enable UI**

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

### Preserve Size

This is used to preserve the values of the local size property group when the object is updated from the library.

### Preserve Commands

This is used to preserve the list of commands when the object is updated from the library.

### Preserve Animations

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Multimedia Viewer

The "Multimedia Viewer" object allows you to display multimedia files such as videos, audios and images at runtime.

The toolbar can be activated /deactivated using the object's properties to allow the user to interact with functions such as volume, reproduction speed and other.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Multimedia Reader Settings

### **Source File**

This is used to enter the complete path of the file to be reproduced.

### **Loading Command**

This is used to set the command to execute when loading the object.

### **Keep Aspect Ratio**

Enabling this property will keep the displayed image unaltered.

### **Loop**

Enabling this property will run the image in a loop so that once the reproduction of image has been terminated it will automatically start from the beginning again.

### **Stretch**

This is used to set which stretch mode to apply to the film/image.

### **Stretch Mode**

This is used to set how to apply the stretch mode to the film/image.

## **Style**

### **Background Color**

This is used to set the container's background color.

### **Text Color**

This is used to set the object's text color.

### **Horizontal Content Alignment**

This is used to horizontally align all the contents within the object.

### **Vertical Content Alignment**

This is used to vertically align all the contents within the object.

### **Disable Anti-Aliasing**

This is used to Enable/Disable the Anti-aliasing function.

## **Fonts**

### **Font Size**

This is used to set the size of the selected font.

### **Font Family**

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more languages defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NEXT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height



This is used to change the object's height.

#### **Width**

This is used to change the object's width.

#### **Border Thickness**

This is used to set the container's border thickness.

#### **Margin**

This is used to set the object's margins.

#### **Lock Object Position**

This locks the object's position on screen.

## **Visibility**

#### **Transparency**

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

#### **Visibility**

This is used to enable the object's visibility.

#### **Visible on Web Client**

This makes the object visible on Web Client.

#### **Visibility Layer Level**

This is used to set the mask that determines the object's visibility.

#### **Zoom Level Visibility X**

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

#### **Zoom Level Visibility Y**

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## **General**

This is a list of the general properties which are the same for objects and draws:

#### **ToolTip**

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### Force as Dynamic on Client

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

### Preserve Size

This is used to preserve the values of the local size property group when the object is updated from the library.

### Preserve Commands

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## URL Streaming Viewer

The URL Streaming Viewer object allows video streaming from a source, such as an IP camera, to be visualized at Runtime using the RTSP protocol.

RTSP (Real Time Streaming Protocol) is a network control protocol for streaming multimedia that uses communication port 554.



The IP Camera can be used to visualize video streaming with other protocols.



The object can be connected to a protected link by setting the required credentials using the appropriate authentication properties.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Media Settings

### Streaming URL

Full media URL to be displayed.

### Stream User

User with whom to connect to the service in streaming.

### Stream Password

User Password with which to use for connecting to the streaming service.

### Force TCP

Enabling this property will force a TCP connection.

### Ignore frames with bad timestamp

When this property is enabled, frame timestamp errors will be ignored while streaming the video.

### High Priority (default = false)

When enabled, this property will give the URL Streaming Viewer object high priority by consuming more resources to provide better video streaming that is more fluid.

### Audio Disabled (default = false)

When this property is enabled, any streaming audio source will be disabled.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### **Position Y**

This is used to set the object's vertical position in pixels.

### **Height**

This is used to change the object's height.

### **Width**

This is used to change the object's width.

### **Border Thickness**

This is used to set the container's border thickness.

### **Margin**

This is used to set the object's margins.

### **Lock Object Position**

This locks the object's position on screen.

## **Visibility**

### **Transparency**

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

### **Visibility**

This is used to enable the object's visibility.

### **Visible on Web Client**

This makes the object visible on Web Client.

### **Visibility Layer Level**

This is used to set the mask that determines the object's visibility.

### **Zoom Level Visibility X**

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### **Zoom Level Visibility Y**

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## **General**

This is a list of the general properties which are the same for objects and draws:



### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### Force as Dynamic on Client

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

### Preserve Size

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## 2.19. Draws

The Movicon NExT toolbox has a category containing basic shapes called "Draws".

The Movicon 'Draws' category is a collection of primitive shapes that can be used by the user to geometrically configure his or her graphical interface as pleased by using the Graphics Editor. In addition, the user can import their own vectorial drawings using their preferred graphics editor providing that the drawings are defined with the XAML format as required by the WPF technology.



To add new points in a drawing please see the paragraph on "Changing Polygons Points" from the "Graphics Editor" section.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Element Settings

### X Start Point

This is used to set the line's start coordinates for the X axis.

### Y Start Point

This is used to set the line's start coordinates for the Y axis.

### X End Point

This is used to set the line's end coordinate for the X axis.

### Y End Point

This is used to set the line's end coordinate for the Y axis.

### Stroke Thickness

Sets the border thickness.

### Stroke Start Line Cap

This is used to set the object's start line cap type (Flat, Square, Round, Triangle).

### Stroke End Line Cap

This is used to set the object's end line cap type (Flat, Square, Round, Triangle).

### Stroke Dash Cap

This is used to set which stroke dash cap type to use for the drawing (Solid dash array types are not managed).

### Stroke Line Join

This is used to set the which format to use at the intersection of two drawing segments (Miter, Bevel, Round).

### Stroke Miter Limit

This is used to set the maximum length of the corner tip when the corner formed by two segments is acute (functions only with the Stroke Line Join = miter property).

### Stroke Dash Offset

This is used to set the distance of the stroke dash from the drawing's initial start point.

### Stroke Dash Array

This is used to set the drawing's line type. In addition to the options shown on the drop down menu, the user can also set the displayed line type by using an array of values (eg. 1,2,1,3 etc.). In this case, the first value in the array, which corresponds to the 0 index, specifies the dash length, the second value corresponding to the 1 index specifies the space in between, the third specifies the dash length again and so on.

#### Point Count

This property is available for the "**Regular Polygon**" and "**Star**" draws and is used to set the number of points to be used by the drawing in question.

#### Inner Radius

This property is available for the "**Regular Polygon**" and "**Star**" draws and is used to set the inner radius of all the angles.

#### Start Angle

This is available for the "**Arc**" and "**Pie**" draws and is used to set the start angle.

#### End Angle

This is available for the "**Arc**" and "**Pie**" draws and is used to set the end angle.

#### Arc Thickness

This is available for the "**Arc**" and "**Pie**" draws and is used to set the border thickness.

#### Stroke Thickness

This is available for the "**Arc**" and "**Pie**" draws and is used to set the border thickness.

#### Arrowhead Angle

This is available for the "**Arrow**" drawing and is used to set the angle of the arrowhead.

#### Arrow Body Size

This is available for the "**Arrow**" drawing and is used to set the size of the arrow's body in proportion to its head.

#### Arrow Orientation

This is available for the "**Arrow**" drawing and is used to set which direction the arrow must face.

#### Anchor Point

This is available for the "**Callout**" object and is a read only property.

#### Callout Style

This is available for the "**Callout**" object and is used to set the object's style.

#### Shades Nr.

This is available for the "**Pipeline**" object is is used to set the number of shades that appear along the pipeline.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.



Some of the previously listed Style properties are only available for draw objects that support them, such as the 'Callout' object.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed.



The Font properties are only available for the draw objects that provide the use of characters such as the 'Callout' object for example.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.



### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### **TAB Stop**

This makes the object selectable at runtime using the TAB button.

### **Text Direction**

This allows you to set the direction of the text within the object.

### **Enable UI**

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

#### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

#### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

#### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

#### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

#### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

#### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## **Animations**

#### **Animations**

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## **Commands**

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## 2.20. Potentiometers

The MoviconNExT potentiometers are configurable objects that enable specified variable values to be displayed or set graphically. These objects are available in the "Analog Commands" section in the toolbox.

The Potentiometers consist of a knob (cursor) and the relative Scale which can be both configured by using their associated properties.

### Tag

#### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

#### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

### Indicator - Basic Aspect

#### Potentiometer Style

This is used to set the potentiometer style.

#### Potentiometer Background Color

This is used to set the container's background color.

#### Outer Ring Background Color

This is used to set the color of the outer ring area.

#### Inner Ring background Color

This is used to set the color of the area outside the ring.

### Engineering Unit

This is used to set the name of the EU to link to the object.

### Numeric Value Format (Advanced)

This is used to set the numeric value format by defining the number of digits and decimal points.

## Indicator Scale

### Minimum Value

This is used to set the allowed minimum value.

### Maximum Value

This is used to set the allowed maximum value.

### Tag Min. Value

This replaces the minimum scale value with the linked variable value.

### Tag Max Value

This replace the maximum scale value with the linked variable value.

### Value Entry Limits for Controls

The EditDisplay, Gauge, LinearMeter, Slider and Potentiometer control types allow tag values to be displayed and changed. When entering values, you will be able to define the Static Limits (defined in Design mode) or Dynamic Limits (linking two runtime editable tags). In addition, these limits can also be defined in the same Tag by means of the link to an Engineering Unit Tag. The priorities with which these limits are used to intervene depend on certain settings. Below are listed the cases in which these limits may intervene:

1. The Tag is not linked to an Engineering Unit. In this situation, only the Static Limits or the Dynamic Limits are considered. The Dynamic Limits always have priority over Static Limits, therefore, if Tags have been insert to manage the Dynamic Limits in the object, these will be used instead of the Static Limits at Runtime.
2. The Tag is linked to an Engineering Unit and has been set with Static Limits and not Dynamic Limits. In this case, the control's "Enable Engineering Unit" property comes into play. When the property is enabled, the control will inherit the Tag's Engineering Unit's limits and will only use those. Conversely, if the control's "Enable Engineering Unit" property is disabled, the most restrictive limits (both the minimum and maximum) will be considered between the Engineering Units and the Static Limits. For instance, supposing that the Engineering Unit limits are min=0 and max=100 and that the control's Static Limits are min=10 and max=200, the control will then use min=10 and max=100 as limits (the maximum of the minimums and the minimum of the maximums).
3. The Tag has been linked with a Engineering Unit and set with Dynamic Limits. In this case, the most restrictive limits will always be considered (both minimum and maximum) between the Engineering Unit and the Dynamic Limits. For instance, supposing the Engineering Units are min=10 and max=100 and that the control's Dynamic Limits are min=5 and max=90, the control will then use

4. min=10 and max=90 as limits (the maximum of the minimums and the minimum of the maximums).

## Indicator Marker

### Marker Color

This is used to set the Marker's back color.

### Marker Border Color

This is used to set the color of the Marker's border color.

### Marker Offset

This is used to set the space between the value marker and the object's border.

### Marker Dimension

This is used to set the value marker's size.

## Indicator Tickmarks

### Tickmark Style

This is used to set the object's tickmark style.

### Major Tickmark Color

This is used to set the color of the major tickmarks.

### Major Tickmark Count

This is used to set the number of major tickmarks on the scale.

### Major Tickmark Length

This is used to set the length of the object's major scale tickmarks.

### Minor Tickmark Color

This is used to set the color of the minor tickmarks.

### Minor Tickmark Count

This is used to set the number of minor tickmarks on the scale.

### Minor Tickmark Length

This is used to set the length of the object's minor scale tickmarks.

### Major Tickmark Thickness (Advanced)

This is used to set the major tickmark's thickness.

### **Major Tickmark Offset (Advanced)**

This is used to set the space of the major tickmarks from the object's border.

### **Show First Major Tickmark (Advanced)**

Show/Hides the scale's first major tickmark.

### **Show Last Major Tickmark (Advanced)**

Show/Hides the scale's last major tickmark.

### **Minor Tickmark Offset (Advanced)**

This is used to set the distance between the minor scale tickmarks and the object's border.

### **Minor Tickmark Thickness (Advanced)**

This is used to set the thickness of the minor tickmarks.

### **Show Minor Tickmarks on Major Tickmarks (Advanced)**

This is used to show minor tickmarks in the same position of the major tickmarks.

## **Indicator Labels**

### **Label Color**

This is used to set the color of the scale's label.

### **Value Label Color**

This is used to set the color of the value label.

### **Engineering Unit Label Color**

This is used to select the color of the engineering unit's label text.

### **Show Labels**

Enabling this will show the labels on the value scale.

### **Show First Label (Advanced)**

Enabling this will show the first scale's label.

### **Show Last Label (Advanced)**

Enabling this will show the last scale's label.

### **Label Text Format (Advanced)**

This is used to set the label's text format that for default is {0:0}. If you wish to keep 2 digits after the colon, for example, this field should be set with {0:0.00}

### **Label Orientation (Advanced)**

This is used to set the orientation of the scale's label.





This property is not supported in WebHMI and WebClient (SVG/HTML5)

### Label Offset (Advanced)

This is used to set the distance between the scale's labels and the border.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more languages defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

### Label Font

This is used to set the label's text font.

### Value Label Font

This is used to set the current value's text font.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### **Menu with Left button**

This enables the contextual menu to show by also using the left mouse key.

### **TAB Stop**

This makes the object selectable at runtime using the TAB button.

### **Text Direction**

This allows you to set the direction of the text within the object.

### **Enable UI**

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## **Animations**

### **Animations**

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Potentiometer Knob

### Min. Value

This is used to set the minimum value allowed.

### Max. Value

This is used to set the maximum value allowed.

### Show Label

Show/Hides the value labels on the scale.

### Show First Label

This is used to enable/disable the showing of the first label on the scale.

### Show Last Label

This is used to enable/disable the showing of the last label on the scale.

### Value Format

This is used to set the numerical value format by defining the number of digits and decimal separators.

### Label Format

This is used to set a format with which to represent values on the labels.

### Label Offset

This is used to set the distance between the labels and the central cursor.

### Knob Radius

This is used to set the size of the potentiometer knob.

### Use Engineering Unit

This is used to inherit the minimum and maximum limits from the engineering unit relating to the object's associated Tag.

## 2.21. Recipe Viewer

The Recipe Viewer is used to manage Recipe operativity at Runtime. It reports the recipe Data Values in table format and provides commands to manage the recipe in the toolbar at the bottom.

### Recipe Selection Combo-Box

There is a combo-box at the top of the object that allows you to:

- Define a new Recipe index to store on database..
- Select (and therefore call) a Recipe index previously saved on database.

### Data Value Grid

There is a grid in the centre part of the object that reports a Recipe Field on each row.

According to the configuration of the 'Editing' property group of each Data Value, a Display or Checkbox may be used in the Value column for data inputs.

### Load Button

This command loads the Recipe index from the Database displaying the values in the grid.

### Save Button

This command takes the values in the grid and saves them on database in the Recipe index selected in the combo-box.

### New Button

This command is used to create a new Recipe index with the values currently displayed in the grid.

### Remove Button

This command is used to remove the index of the currently selected Recipe from the Database.

### Import Button

This command is used to import a Recipe index from a ".csv" file whose structure must be coherent with that of the recipe.

### Export Button

This command exports the Recipe Index selected in the combo-box in a ".csv" file.

### Read Button

This button is used to execute a read of the device's data (PLC, ecc.) to update the active Recipe index fields.

After which a new Recipe can be saved with a different index or the currently active one can be saved with the 'Save' command.

### Write Button

This command is used to execute a write of the Recipe index data to the device (PLC, etc.).





The values written on the PLC are always those present on the Database. This means that in order to write a Recipe's changed values on the device, these value will first need to be saved on database using the Save command.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Recipe Viewer

### Recipe Name

This is used to assign a name to the recipe.

### Graphic Buttons

This is used to replace the text in buttons with icons.

### Show Button

Enabling any one of these properties will show the corresponding button in the recipe viewer.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more languages defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NEXT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### Force as Dynamic on Client

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

#### Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

#### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

#### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

#### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Execution

#### Group by Name

This is used to group recipe values by name.

#### Use PAD

Enabling this option will activate the numeric pad for digiting values (indispensable for touch screen systems).

#### Show Search Box

This is used to show the Search Box to search data with.

#### Read/Write Timeout

This is used to define a timeout during the read/write of data.

## 2.22. Scheduler Viewer

The 'Scheduler Viewer' is used to show or edit the project's scheduler resources at runtime. The scheduler resources that are displayed by the Scheduler Viewer object are those that have been enabled with the 'Runtime Editable' in design mode. Those schedulers that have not been enabled with this option will not be displayed or configurable at runtime.

### Tag

#### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

#### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

### Scheduler Settings

#### Connection String

This is used to set a reference data source connection string for the object.

#### Startup Scheduler

This is used to set the scheduler assigned as startup scheduler.

#### Time Scale

This is used to set the time scale for the workview.

#### WorkView Start Time

This is used to set the start time for the work view displayed in the weekly settings pane when the "Show Workview Time Only" is active.

#### WorkView End Time

This is used to set an end time for the workview displayed in weekly settings pane when the "Show Workview Time Only" is active.

#### Show Workview Time Only

Enabling this option will display the workview only in the weeking settings pane.

#### User Based Runtime Settings



Enabling this option will customize the object's style based on user at runtime.

#### **Write access level for editing object at runtime**

This is used to set a minimum user level to allow access to edit the object at runtime.

#### **Access mask for editing object at runtime**

This is used to set a user access mask to allow the object to be edited at runtime.

## **Style**

#### **Background Color**

This is used to set the container's background color.

#### **Text Color**

This is used to set the object's text color.

#### **Horizontal Content Alignment**

This is used to horizontally align all the contents within the object.

#### **Vertical Content Alignment**

This is used to vertically align all the contents within the object.

#### **Disable Anti-Aliasing**

This is used to Enable/Disable the Anti-aliasing function.

#### **Weekly Event Color**

This is used to set a color for weekly events.

## **Fonts**

#### **Font Size**

This is used to set the size of the selected font.

#### **Font Family**

This is used to set a font type to use for text.

#### **Font Style**

This is used to set the style for the select font. The options are:

- Normal
- Oblique

- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more languages defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### **Is Interactive**

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### **Menu with Left button**

This enables the contextual menu to show by also using the left mouse key.

### **TAB Stop**

This makes the object selectable at runtime using the TAB button.

### **Text Direction**

This allows you to set the direction of the text within the object.

### **Enable UI**

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

### Preserve Size

This is used to preserve the values of the local size property group when the object is updated from the library.

### Preserve Commands

This is used to preserve the list of commands when the object is updated from the library.

### Preserve Animations

This is used to preserve the list of animations when the object is updated from the library.

### Preserve Menu

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## **Animations**

### **Animations**

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## 2.23. Selectors

The Movicon.NEXt Switches and Selectors are used for executing toggle commands or setting project variables.

These elements are accessible through the "Buttons Switches" group in the toolbox and a major part of their functions and properties resemble those belonging to the checkboxes. Like the objects from the "Buttons-Checkbox" group, they can be set with tri-state mechanics to assume three different positions. The tri-state mechanics enables the set variable to assume a new value each time the switch or selector area is clicked on. Therefore with each click the variable passes from the zero value to value one then value two and then back to the zero value.

The switch's/selector's image changes in function with the new variable value, alternating between three possible images: **switch/selector in zero position**, **switch/selector in position one** and **switch/selector in position two**. The variable can also be set by keeping the mouse button pressed down on the selector and moving it to another position. In this case the variable will be set to the new value only when the mouse button is released and the switch/selector is in a different position from the one it started with.



To exploit the tri-state animation to the full use the switched from the "Selector 3P" sub class which are graphically preset for this type of use.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Button Settings

### Click Mode

This is used to set the click type to generate command execution (release, press, hover).

### Is Three State

This is used to set the button as a three state button if button is enable with this option (only available for Checkboxes).

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:



- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more languages defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NEXT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

#### Border Thickness

This is used to set the container's border thickness.

#### Margin

This is used to set the object's margins.

#### Lock Object Position

This locks the object's position on screen.

## Visibility

#### Transparency

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

#### Visibility

This is used to enable the object's visibility.

#### Visible on Web Client

This makes the object visible on Web Client.

#### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

#### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

#### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

#### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

#### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

#### **Is Interactive**

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

#### **Menu with Left button**

This enables the contextual menu to show by also using the left mouse key.

#### **TAB Stop**

This makes the object selectable at runtime using the TAB button.

#### **Text Direction**

This allows you to set the direction of the text within the object.

#### **Enable UI**

This is enabled for default to allow user interactivity with the object in the UI.

#### **Is Focusable**

Activates / Deactivates focus on the object.

#### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

#### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

#### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

#### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

#### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

#### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

### Preserve Size

This is used to preserve the values of the local size property group when the object is updated from the library.

### Preserve Commands

This is used to preserve the list of commands when the object is updated from the library.

### Preserve Animations

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

### **Configurable Write Access Level**

This is used to set the minimum access level allowed for editing/configuring the object at runtime.

### **Configurable Write Access Mask**

This is used to set the access area to allow the object to be edited/configured at runtime.

## **Animations**

### **Animations**

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## **Commands**

### **Commands**

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## 2.24. Vertical and Horizontal Sliders

The Movicon NExT sliders are configurable sliders that enable valuable values to be displayed or set graphically. The Slider objects are available from the "ToolBox" section in the "Analog Commands" group. The Slider consists of a Scale and Bar both of which can be configured through their properties.

### Tag

#### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

#### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

### Slider - Basic Aspect

#### Slider Style

This is used to select which style to use for the object.

#### Tick Placement

This is used to set where to place scale ticks.

#### Minor Tick Size

This is used to set the tickmark's minor size.

#### Tick Size

This is used to set the size of the scale's tickmarks.

#### Tick Thickness

This is used to set the scale's tickmark thickness.

#### Show Scale Labels

Enabling this will show the value labels on the scale.

### Use Engineering Unit

Enabling this option will allow the object to use the engineering unit and inherit the scale value limits.

### Major Tick Frequency

This is used to set the frequency with which the major tickmarks appear on the scale.

### Invert Direction

This is used to invert the direction of the slider.

### Rep. Delay Time

This is used to set the delay time before repeating the command again.

### Rep. Interval Time

This is used to set the delay time with which the command will be repeated.

### Tooltip Digit Details

This is used to set the number of decimal digits.

### Snap To Tick

When enabled snaps the slider cursor to scale tick nearest to the point in which the user has clicked to change the value.

### Tick Frequency

This is used to set the scale's tick frequency.

### Move to Point

When enabled will force the variable value to the point where the cursor is upon clicking.

## Slider - Scale

### Minimum Value

This is used to set the allowed minimum value.

### Maximum Value

This is used to set the allowed maximum value.

### Tag Min. Value

This replaces the minimum scale value with the linked variable value.

### Tag Max Value

This replace the maximum scale value with the linked variable value.

### Value Entry Limits for Controls

The EditDisplay, Gauge, LinearMeter, Slider and Potentiometer control types allow tag values to be displayed and changed. When entering values, you will be able to define the Static Limits (defined in

Design mode) or Dynamic Limits (linking two runtime editable tags). In addition, these limits can also be defined in the same Tag by means of the link to an Engineering Unit Tag. The priorities with which these limits are used to intervene depend on certain settings. Below are listed the cases in which these limits may intervene:

1. The Tag is not linked to an Engineering Unit. In this situation, only the Static Limits or the Dynamic Limits are considered. The Dynamic Limits always have priority over Static Limits, therefore, if Tags have been insert to manage the Dynamic Limits in the object, these will be used instead of the Static Limits at Runtime.
2. The Tag is linked to an Engineering Unit and has been set with Static Limits and not Dynamic Limits. In this case, the control's "Enable Engineering Unit" property comes into play. When the property is enabled, the control will inherit the Tag's Engineering Unit's limits and will only use those. Conversely, if the control's "Enable Engineering Unit" property is disabled, the most restrictive limits (both the minimum and maximum) will be considered between the Engineering Units and the Static Limits. For instance, supposing that the Engineering Unit limits are min=0 and max=100 and that the control's Static Limits are min=10 and max=200, the control will then use min=10 and max=100 as limits (the maximum of the minimums and the minimum of the maximums).
3. The Tag has been linked with a Engineering Unit and set with Dynamic Limits. In this case, the most restrictive limits will always be considered (both minimum and maximum) between the Engineering Unit and the Dynamic Limits. For instance, supposing the Engineering Units are min=10 and max=100 and that the control's Dynamic Limits are min=5 and max=90, the control will then use
4. min=10 and max=90 as limits (the maximum of the minimums and the minimum of the maximums).

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

### Tick Color

This is used to set the color of the ticks.

### Decrease Button Color

This is used to differentiate the slider object's bar color according to the value of the tag assigned to the object.



### Thumb Color

This is used to set the color of the slider's cursor.

### Bar Color

This is used to set the color of the bar.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### Force as Dynamic on Client

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

#### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

#### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

#### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

#### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

#### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

#### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

#### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

#### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

#### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### Access Level from Tag

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## 2.25. Switches

### Switch Buttons, Lights and Checkboxes

The Movicon NEXt buttons are divided into two types according to their mechanics but they all have the same style properties. Some buttons have ON-OFF mechanics while others execute COMMANDS. Therefore you must insert the button type that is most appropriate for the functionality needed.



Be reminded that the styles of each object you insert on screen can be modified graphically but their mechanics cannot.

### Buttons-Digital Checkboxes

In Toolbox section you will find a vast range of buttons by type and style. The main characteristic of this category is the possibility to associate a variable for toggling between 0 and 1 (On-Off command). Even though this object's styles are the same as those of the command buttons, its "Check-Box" mechanics require you to associate it a Tag variable. This can be done by using the Drag & Drop techniques or by associating the Tag using the properties or quick select menu.

## CheckBox

The checkBox is a classical two state selector button. A Movicon NExT variable can be associated to this control and set to the "0" value (unchecked box) or the "1" value (checked box) according to the state of the box.

The state can be selected with a mouse click or using the keyboard to select the by selecting the component with the TAB or MAIUSC+TAB keys and pressing ENTER.

## Lamps

Lamps are luminous indicators that are used to represent the state of the connected variable. This is done by using the behaviour properties to set them to display based on two or three state mechanics.

These objects are read only for default but it is also possible to control the assigned variables through their associated properties.

## Toggle Switches

The MoviconNExT Switches and Selectors are used to execute toggle commands or set project variables. These elements are accessible through the "Buttons Switches" group in the toolbox and a major part of their functions and properties resemble those belonging to the checkboxes. As with the objects from the "Buttons-Checkbox" group, they can be set with tri-state mechanics to assume three different positions. The tri-state mechanics enable the set variable to be modified to assume a new value each time the switch or selector area is clicked on. Therefore with each click the variable passes from the zero value to value one then value two and then back to the zero value.

The switch/selector's image changes in function with the new variable value, alternating between three possible images: **switch/selector in zero position**, **switch/selector in position one** and **switch/selector in position two**. The variable can also be set by keeping the mouse button pressed down on the selector and moving it to another position. In this case the variable will be set to the new value only when the mouse button is released and the switch/selector is in a different position from the one it started with.



To exploit the tri-state animation to the full, use the switches from the "Selector 3P" sub class which are graphically preset for this type of use.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Button Settings

### Click Mode

This is used to set the click mode to generate the command execution (release, press, hover).

### Is Three State

If provided, this option is used to set the button as three-state (only available for Checkboxes).

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

### Back Color When ON

This is used to set the object's back color which will change when object is active.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic



## Font Weight

This is used to select the thickness of the selected font.

## Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more languages defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

# Layout

## Position X

This is used to set the object's horizontal position in pixels.

## Position Y

This is used to set the object's vertical position in pixels.

## Height

This is used to change the object's height.

## Width

This is used to change the object's width.

## Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

#### **Menu with Left button**

This enables the contextual menu to show by also using the left mouse key.

#### **TAB Stop**

This makes the object selectable at runtime using the TAB button.

#### **Text Direction**

This allows you to set the direction of the text within the object.

#### **Enable UI**

This is enabled for default to allow user interactivity with the object in the UI.

#### **Is Focusable**

Activates / Deactivates focus on the object.

#### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

#### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

#### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

#### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

#### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

#### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

#### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

### Preserve Size

This is used to preserve the values of the local size property group when the object is updated from the library.

### Preserve Commands

This is used to preserve the list of commands when the object is updated from the library.

### Preserve Animations

This is used to preserve the list of animations when the object is updated from the library.

### Preserve Menu

This is used to preserve the set contextual menu, when the object is updated from the library.

### Preserve Expression

This is used to preserve the value conversion expression, when the object is updated from the library.

#### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

#### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

#### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

#### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

#### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

#### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

#### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## **Animations**

#### **Animations**

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## 2.26. TAB Screens

Movicon has a component called the "Embedded Screen Tab Selector" which can be inserted within other screens. This object is used for representing various Screens in the project and embedding them into other Screens as vectorial components.

The particular feature of this object is that it displays the designs, components and animations contained in the screen associated to it.

The possibility to create general project layouts with predefined screens will save users a great deal of time when designing engineering projects.

The names of the relative screens are reported in the object's Tabs. These names can be customized and managed with multi-languages using the project's String Table. This is done by simply inserting an ID String with the name and path of the screen to automatically substitute the ID at runtime.

For example, if you insert a screen named 'Machine1' in the 'Line' folder within the EmbeddedTabScreen, you will only need to insert an 'Line/Machine1' ID in the String Table in order to display the text corresponding to the ID String in the EmbeddedTabScreen's Tab at runtime in the language selected.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Embedded Screen Tab

### Object Configuration

See chapter on: Smart Properties

### Scroll Animation

This is used to set the Tab scroll animation type.

### Page Header Alignment

This is used to set the tab's alignment in the object.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more languages defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin



This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### **TAB Stop**

This makes the object selectable at runtime using the TAB button.

### **Text Direction**

This allows you to set the direction of the text within the object.

### **Enable UI**

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

#### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

#### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

#### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

#### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

#### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

#### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## **Animations**

#### **Animations**

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## **Commands**

## Commands

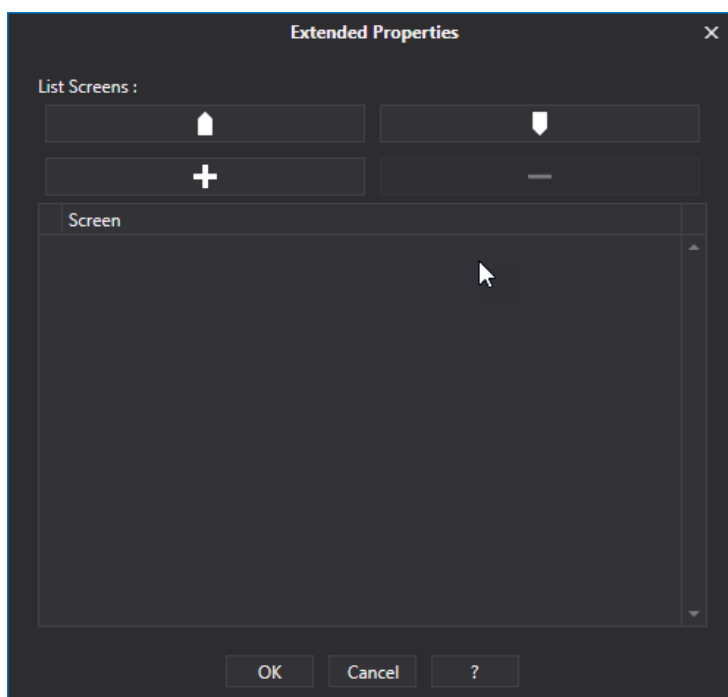
This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Smart Properties



These smart properties allow you to add or remove embedded screens and move them around in the 'tab list' as needed.

## 2.27. Texts

### Texts

The "Text" object is used to set a static text box. The user can also set text within the box and associate a tag to use one of standard animations that are common to all objects.



It is possible to use the following "\n" escape sequence to force the text to go to the beginning of the next line. For example the string:

Item 1\nItem 2\nItem 3

in the visualization phase, for example a text box or button will appear as follows:

- Item 1
- Item 2
- Item 3

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

### **Min. Lines**

This is used to set the minimum number of text lines.

### **Max. Lines**

This is used to set the maximum number of text lines.

### **Enable Horizontal ScrollBar**

Enables the horizontal scroll bar according to which option has been selected.

### **Enable Vertical ScrollBar**

Enables the vertical scroll bar according to which option has been selected.

### **Selected Text Color**

This is used to select a different color for the selected text.

## **Fonts**

### **Font Size**

This is used to set the size of the selected font.

### **Font Family**

This is used to set a font type to use for text.

### **Font Style**

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### **Font Weight**

This is used to select the thickness of the selected font.

### **Set Font based on Language**

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more languages defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one

language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed.

### Text Wrapping

This is used to select whether or whether not to return text to beginning of next line when reaching the text box's border.

### Text Alignment

This is used to set the text alignment within the object.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility



### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### Preserve Text

This is used to preserve the added text string when the object is updated from the library.

### Text

This is used to the object with the Default Text.

## User Access

### Access Level

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### User Access Role

This is used to set the name of the user password group to allow user interaction with this object.

### Access Level from Tag

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Animated Text

The animated text box can be configured by using the "Animated Text" object. The text inside this box and be set by the user and then be associated to a tag taken directly from the string table by using the **Smart TAG** settings located at the end of the object's list of properties.



It is possible to use the following "\n" escape sequence to force the text to go to the beginning of the next line. For example the string:

Item 1\nItem 2\nItem 3

in the visualization phase, for example a text box or button will appear as follows:

- Item 1
- Item 2
- Item 3

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Dynamic Text

### Object Configuration

See chapter below on Smart Properties

### Default Text

This is used to set the Default Text for the object.

### Stretch Type

This is used to select which stretch type to apply the text to the object.

### Animation Time

This is used to set the pause time between one animation and the next.

### Control Render Transform Origin

This is used to define the control render transform origin from which to execute the animation.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more languages defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).



### **Menu with Left button**

This enables the contextual menu to show by also using the left mouse key.

### **TAB Stop**

This makes the object selectable at runtime using the TAB button.

### **Text Direction**

This allows you to set the direction of the text within the object.

### **Enable UI**

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## **Animations**

### **Animations**

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Smart Properties

### Value

Value Threshold to reach for activating animation.

### Text

This is used to set the text to be displayed.

### Execution Type

This is used to set the execution type to assign to the animation.

### Animation Time

This is used to set a scroll time when two or more images have been set in the back image.

### Back Image

This is used to set the object with a background image.

### Stretch

This is used to adapt the image to the side of the container.

### Text Color

This is used to set the color of the text.

### Tag

This field is used to select a Tag whose value will be displayed at runtime to substitute the content in the "Text" column.

The Tag can be string or numeric type. When the Tag is string type and contains a text that corresponds to an "String ID", the change language will also be involved.

## 2.28. Trends

### Trend and Data Analysis Viewers

The Trend is used for graphically representing variable trends using curves and analysis of data filed by data recorder engines. Different features are provided according to the trend object being used.

The Trend has a double function. It displays runtime data and logged data (run-pause). The Data Analysis performs analysis on logged data only and provides more advance features for analysing data.

### Configuring the Pen List in Runtime

This functionality can be used in runtime and allows the user to create and manage settings for pen lists to be displayed in the RealTime Trend and Data Analysis.

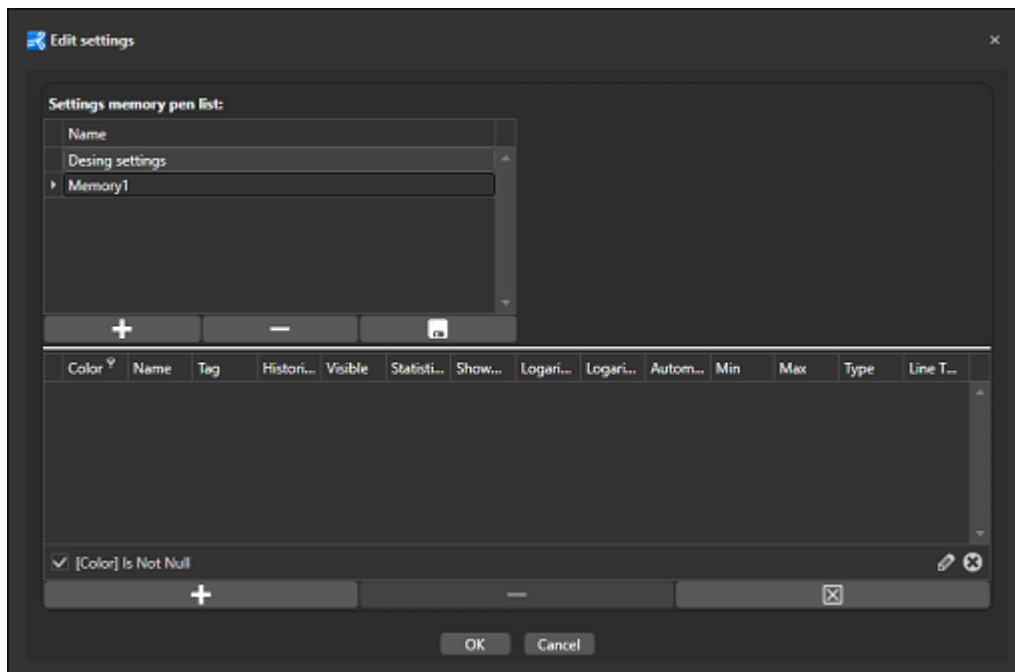
When enabling the 'Allow Runtime Changes' in the Data Analysis/Trend Style property group, a toolbar will display at the top of the object which can be used to access the Pen List's settings editor window.

In addition to displaying the currently active pen configuration in runtime mode, this toolbar also displays two command buttons:

- Edit Settings: when pressing this button a pen list settings editor popup window will appear.
- Reset Settings: when pressing this button the default pen settings, defined by the project designer in development mode, will be called and automatically loaded.

#### Pen List setting editor window

The popup window used for configuring and managing the pen lists will display as shown in the image below.



This window is composed of two main parts: the top part contains the list of currently available settings for the selected object while the part underneath contains the list of pens associated to the settings shown in the above part.

Different lists of settings can be managed by using the 'Add', 'Remove' and 'Save' commands. These lists will be created by cloning the currently selected settings on the list.

Pens can be added and removed in the same way from the edit settings window using the "Add Pen", 'Remove Pen' or 'Clear All'

Access to commands linked to the pen list settings in runtime, can be protected using the project user management: simply set the access level desired using the object's "Configuration Write Access Level" and "Configuration Write Access Mask" style properties.



When the currently logged in user does not have the right access level required, they will not be allowed to edit the existing pen list settings but only activate them.

## Chart

Movicon Next has a window object which can be configured as pleased to create charts in different formats in relation to the Movicon variables.

The charts are built from objects that have different style characteristics compared to the other Movicon objects described so far. These different style types permit more sophisticated and powerful configurations.



The chart object is used for representing the values of different tags by connecting them to pens for which style properties can be configured using the additional property editor.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Chart - Settings

### Object Configuration

See chapter below on Smart Properties

### Sample Number

This is used to set the number of samples to be represented on the curve.

### Linked Pen Name

This is used to set the name of the pen associated to the chart.

### **Push Tag**

This is used to insert the name of the tag for the 'Push' command. When setting this variable to '1', the current value will be shown on the chart.

### **Reset Tag**

This is used to insert the name of the tag for the "Reset" command. Setting this variable to '1' will remove the chart's buffer.

### **Zoom Tag**

This is used to insert the name of the tag for the "Zoom" command. Setting this variable to '1' will activate the zoom.

### **Rotated**

This is used to enable/disable the object's rotation.

### **Enable 3D Style**

This is used to enable the chart to display in 3D.

### **Pie Chart**

This enables the chart to display as a pie chart.

### **Enable Animation**

Enable/Disable the initial curve animation.

### **Animation Mode**

This is used to define the object's animation mode.

### **Resolve Label Overlapping Mode**

This is used to set the label overlapping mode type.

### **3D Zoom Percent**

This is used to set the zoom factor for the chart in 3D.

### **Database Connection Timeout**

This is used to set maximum time to wait in seconds to connect the object to the database. Value left set at '0' equals a timeout of 30 seconds.

## **Chart - Scale**

### **X Axis Title**

This is used to set the title for the X axis.

### **Y Axis Title**

This is used to set the title for the Y axis.

**Min. Scale Value**

This is used to set the minimum value for the Y axis.

**Max. Scale Value**

This is used to set maximum value for the Y axis.

**Automatic Scale**

This is used to enable the automatic scale for the primary axis.

**Logarithmic Y Scale**

This enables/disables the logarithmic scale on the Y axis.

**Logarithmic Y Scale Base**

This is used to set the base for the logarithmic scale on the Y axis (applicable only when the "Logarithmic Y Scale" option has been enabled).

**Use Engin. Unit for Min/Max Values**

Enables/Disables the displaying of the scale's MIN and MAX values by inheriting them from the EU relating to the Tag associated to the pens.

## Chart - Aspect

**Plot Area Background Color**

This is used to set the plot area's background color.

**Trend Area Background Color**

This is used to set the trend area's background color.

**XY Title Font Color**

This is used to set the color of the object's title.

**Axis Value Color**

This is used to set the color of the axis' labels.

**Show Value Labels on plot**

This is used to enable/disable the display of reference value labels on the plot.

**Show Legend Area**

This is used to enable/disable the legend area within the object.

**Auto Hide Toolbar**

Enabling this will automatically hide the toolbar at the top of the object.



### **Legend Margins**

This is used to set the legend area's margin values.

### **Horizontal Legend Position**

This is used to set the legend area's horizontal position.

### **Vertical Legend Position**

This is used to set the legend area's vertical position.

### **Max. Legend Height**

This is used to set the maximum height for the legend area.

### **No. Decimal Digits on Labels**

This is used to set the number of decimal digits to show on the pie labels for floating point values.

### **Show Percent Values on Pie Chart**

Enabling this will show the percent values of points on the Pie Chart.

## **Style**

### **Background Color**

This is used to set the container's background color.

### **Text Color**

This is used to set the object's text color.

### **Horizontal Content Alignment**

This is used to horizontally align all the contents within the object.

### **Vertical Content Alignment**

This is used to vertically align all the contents within the object.

### **Disable Anti-Aliasing**

This is used to Enable/Disable the Anti-aliasing function.

## **Fonts**

### **Font Size**

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NEXT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed.

### XY Title Font Style

This is used to set the object's title font style.

### XY Axis Font Style

This is used to set the axes' title font style.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

#### **Position Y**

This is used to set the object's vertical position in pixels.

#### **Height**

This is used to change the object's height.

#### **Width**

This is used to change the object's width.

#### **Border Thickness**

This is used to set the container's border thickness.

#### **Margin**

This is used to set the object's margins.

#### **Lock Object Position**

This locks the object's position on screen.

## **Visibility**

#### **Transparency**

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

#### **Visibility**

This is used to enable the object's visibility.

#### **Visible on Web Client**

This makes the object visible on Web Client.

#### **Visibility Layer Level**

This is used to set the mask that determines the object's visibility.

#### **Zoom Level Visibility X**

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

#### **Zoom Level Visibility Y**

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## **General**

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be

executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### Force as Dynamic on Client

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

### Preserve Size

This is used to preserve the values of the local size property group when the object is updated from the library.

#### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

#### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

#### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

#### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

#### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

#### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

#### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

#### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

#### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

Animations

Animations


This property gives access to the window containing dynamic animation commands that can be associated to the object.

Commands

Commands

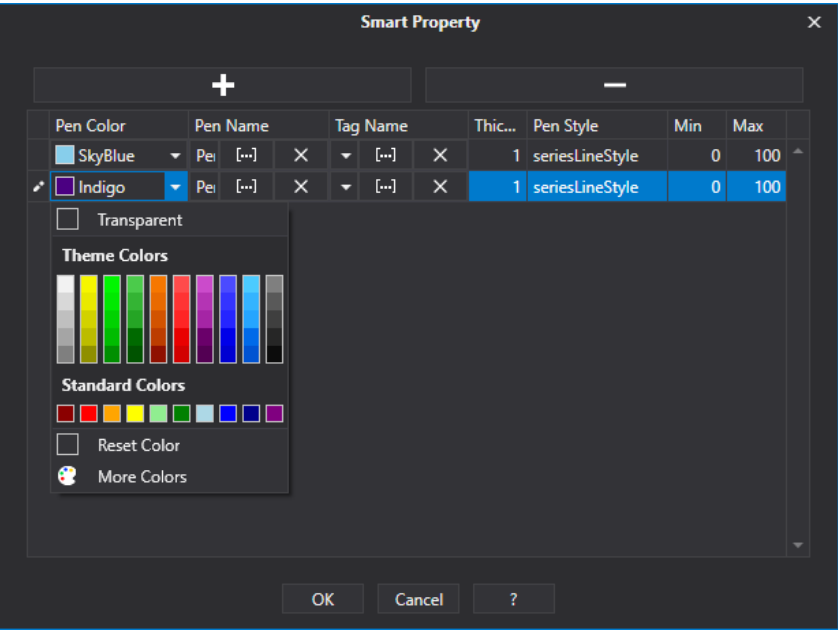
This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

Smart Properties



### Pen Color

This is used to select a pen color.

### Pen Name

This field is used for entering a name with which the pen will be displayed in the legend area. It is also possible to insert a String ID to manage language changes at runtime.

### Tag Name

This field is used for selecting the variable to associated to the pen (or the DataLogger column in cases when the variable is associated to a DataLogger and not a Historian).

### Thickness

This field is used to the thickness of the pen line.

### Pen Style

This field is used for selecting a style with which to display the pen.

### Min/Max

These two values define the minimum and maximum limits for displaying the pen with. In case where the Tag associated to the pen has been associated with an Engineering Unit, the minimum and maximum values will be taken directly from the Engineering Unit and these two field will be ignored.

## XY Chart

The XY Chart is one of many graphical objects provided by Movicon.NExT to represent TAG values which have been programmed in the project.

Each point, called PENS, presented in the Chart XY graphics has a X coordinate associated to the value of a TAG and a Y coordinate associated to the value of another TAG.

When the values of the 2 tags change, the point will move in the chart until the Force Value command in the object's Toolbar is used to fix its positions.

When this is done several times and when the values of associated tags change, the chart will eventually populate with points until the maximum value, set by the programmer the configuration phase, has been reached (Sample Number parameter).

The following 4 commands in the ChartXY's ToolBar are used to:

- Reset the displayed samples
- Run samples
- Display labels containing values of the sampled points
- Display the legend containing the names of displayed pens

Several pens can be displayed on a Chart XY at the same time. The maximum limit is determined by the user's ability to read the chart.



Arrays can also be presented on the Chart XY: numeric array values can be associated to the x and y coordinates of a pen and played on the chart xy. When arrays are used, it is not necessary to populate the chart with successive samples of the values associated to the pens as all the values contained in the arrays are presented instantaneously.



The following shortcuts can be used to zoom in on a portion of the graph:

- Hold down the SHIFT key, then select the area with the mouse and use the Drag & Drop techniques to enlarge it to the desired size.
- Use the SHIFT + CLICK to zoom in on the cursor's position.
- Use ALT + CLICK to restore the zoom previously set.
- Use CTRL + "+" or "-" to zoom (or reduce to return to the maximum original size) by a progressive 20%
- Use CTRL + "Z" to undo the last modifications made in the zoomed area.



The mouse's "**wheel button**" only zooms along the abscissa axis, while the pinch-to-zoom on touch screens will allow you to zoom both along the abscissa axis and along the ordinate axis.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## XY Chart - Settings

### Object Configuration

See chapter below on Smart Properties

### Sample Number

This is used to set the number of samples to be represented in the chart.

### Linked Pen Name

This is used to set the name of the pen associated to the chart.

### Push Tag

This is used to associate a Tag to enter a new value on the chart.

**Reset Tag**

This is used to insert the name of the tag for the command to reset the sampling values on the chart.

**Rotated**

This is used to enable/disable the object's rotation.

**Auto Hide Toolbar**

Enabling this will automatically hide the toolbar at the top of the object.

**Enable Animation**

Enable/Disable the initial curve animation.

**Animation Mode**

This is used to define the object's animation mode.

**Resolve Label Overlapping Mode**

This is used to set the label overlapping mode type.

**Database Connection Timeout**

This is used to set maximum time to wait in seconds to connect the object to the database. Value left set at '0' equals a timeout of 30 seconds.

## **XY Chart - Scale**

**X Axis Title**

This is used to set the title for the X axis.

**Y Axis Title**

This is used to set the title for the Y axis.

**Min. Scale Value**

This is used to set the minimum value for the Y axis.

**Max. Scale Value**

This is used to set maximum value for the Y axis.

**Automatic Scale**

This is used to enable the automatic scale for the primary axis.

**Logarithmic Y Scale**

This enables/disables the logarithmic scale on the Y axis.

### **Logarithmic Y Scale Base**

This is used to set the base for the logarithmic scale on the Y axis (applicable only when the "Logarithmic Y Scale" option has been enabled).

### **Use Engineering Unit**

This enables the use of the EU values for the Y Scale's Min and Max values.

### **Scale Padding Factor**

This is used to set the space of the margins between the border and chart's scales.

## **XY Chart - Aspect**

### **Plot Area Background Color**

This is used to set the plot area's background color.

### **Show Marker**

This property is enabled for default and allows markers to be displayed for all the curves that have been inserted in the object. When disabling this property, the markers will not be shown for all the curves.

### **Trend Area Background Color**

This is used to set the trend area's background color.

### **XY Title Font Color**

This is used to set the color of the object's title.

### **Axis Label Foreground**

This is used to set the color of the axis' labels.

### **Show Value Labels in Chart Area**

This is used to enable/disable the display of reference value labels on the plot.

### **Show Legend Area**

This is used to enable/disable the legend area within the object.

### **Legend Margins**

The is used to set the legend area's margin values.

### **Horizontal Legend Position**

This is used to set the legend area's horizontal position.

### **Vertical Legend Position**

This is used to set the legend area's vertical position.

# Style

## Background Color

This is used to set the container's background color.

## Text Color

This is used to set the object's text color.

## Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

## Vertical Content Alignment

This is used to vertically align all the contents within the object.

## Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

# Fonts

## Font Size

This is used to set the size of the selected font.

## Font Family

This is used to set a font type to use for text.

## Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

## Font Weight

This is used to select the thickness of the selected font.

## Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style

- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

### XY Axis Font Style

This is used to set the font of the title of the object's axes.

### XY Title Font Style

This is used to set the font of the Object's title.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### **Text Direction**

This allows you to set the direction of the text within the object.

### **Enable UI**

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**



This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

#### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

#### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

#### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

#### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

#### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## **Animations**

#### **Animations**

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## **Commands**

#### **Commands**

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Smart Properties

### Pen Color

Indicates the pen color.

### Point Label Color

This is used to set the color of the label for the selected point.

### Pen Name

This is used to enter the name with which the pen will be displayed in the legend. It is also possible to enter a String ID to manage language changes at runtime.

### X Tag

This field is used to select the variable to associate to the X axis.

### Y Tag

This field is used to select the variable to associate to the X axis.

### Thickness

This field is used to the thickness of the pen line.

### Min/Max

These two values define the minimum and maximum limits for displaying the pen with. In case where the Tag associated to the pen has been associated with an Engineering Unit, the minimum and maximum values will be taken directly from the Engineering Unit and these two field will be ignored.

## Data Analysis RT

The Data Analysis can be used to display both real-time and Historian data.

It has two "RunStop" and "OnlyStop" operating modes. When selecting the first mode, the values will be displayed in real-time mode. Selecting the second one will allow you to carry out an in-depth analysis on Historian data.

The Data Analysis object can be defined with a list of pens to represent data graphically both in runtime (made possible by using the **smart tags** property bar) and in development mode using the appropriate buttons displayed in the object's 'legend' window.



The following shortcuts can be used to zoom a portion of the graph:

- keep the SHIFT pressed down, select the area with the mouse and Drag&Drop until you reached the enlarged zoom size desired
- use SHIFT+CLICK to zoom the cursor position
- use ALT+CLICK to restore zoom back to previous setting.



If you zoom in with the mouse wheel, it will only work along the axis of the abscissas. Instead the pinch-to-zoom used in touch screens works along both the axis of the abscissas and the axis of the ordinates.

The Data Analysis also provides tools for representing the analysis and comparisons of data deriving from the DB of the associated Historian. These tools are made available to the user in runtime in the Toolbar at the top of the object.



The buttons are described below in the order they are shown on the Toolbar:

- **Designer:** permits access to the chart designer to customize the window's graphical appearance.
- **Refresh:** Reloads data based on the current time (currently set).
- **Show or Hide cross hair cursor:** Shows or hides the cross hair cursor which appears in correspondence to the sampled values.
- **Show or Hide labels:** Shows or hides the fixed labels that appear in correspondence to the sampled values.
- **Expand or Collapse:** Removes the toolbar by expanding the graph in the screen area.
- **Previous/Next:** Displays the previous or next page relating to the time range selected in the box at the side.
- **Show all Value:** loads all the sampled data.
- **Show Last Minute Value:** Sets the reference base time (minutes) for representing data in graphs.
- **Show Last Hour Value:** Sets the reference base time (hours) for representing data in graphs.
- **Show Last Day Value:** Sets the reference base time (days) for representing data in graphs.
- **Show Last Week Value:** Sets the reference base time (weeks) for representing data in graphs.
- **Show Last Month Value:** Sets the reference base time (months) for representing data in graphs.
- **Show Last Year Value:** Sets the reference base time (years) for representing data in graphs.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Data Analysis - Settings

### Object Configuration

Please see the section below on Smart Properties

### Layout Configuration

This is used to configure the layout with which the object will be presented by modifying the legend and other pop-up windows.

### Runtime Mode

This is used to set the DA's runtime mode.

### Record Every

This is used to set when to record.

### Show Time Frame

This is used to display the time frame within which data can be viewed in the graph's area in RUN mode.

### Max Records

This is used to set the maximum number of records that can be obtained from the DB.

### Filter Type

This is used to set a time filter for loading data from the database. The possible options are:

- All
- Minute
- Hour
- Day
- Week
- Month
- Year

### DB Connection String

This is used to set the DB connection string (the one for default will be used when left blank).

### Max. No. Database Access Retry Errors

This is used to set a maximum number of retry errors when attempting to read data from the database. In the event of a Deadlock during the data acquisition phase, you will need to increase the value of this property bearing in mind that it is likely to increase as the following factors increase: the number of samples set in the DataAnalysis, the size of the data table from which data is read, the requested time range and the number of pens used. The default value for this property is 1.

### Use Aggregation

This is used to enable/disable the the aggregation of data fetched from the DB.

### Max Aggregation Factor

This is used to set the aggregation factor of data fetched from the DB.

### Date/Time Format

This property allows you to use a string to change the object's Date/Time style.

- **Standard format strings for Date/Time values**

The standard Date/Time format strings contain one character. This character defines the model used for representing the value (and in which mode to display the number of years, months and so on).

Format Specifier	Description	Result of formatting a sample value (March 25 2019 12:00:00)
d	Short date pattern	3/25/2019
D	Long date pattern	Monday, March 25, 2019
t	Short time pattern	12:00 AM
T	Long time pattern	12:00:00 AM
f	Full date/time pattern	Monday, March 25, 2003 12:00 AM
F	Full date/time pattern	Monday, March 25, 2003 12:00:00 AM
g	General date/time pattern	3/25/2019 12:00 AM
G	General date/time pattern	3/25/2019 12:00:00 AM

- **Custom format Strings for Date/Time**

To create models for Date/Time values, you will need to combine the strings listed in the table below. These strings represent the year, month, day, number and so on in different formats.

Symbol	Meaning	Result of formatting a sample value (March 25 2019 12:00:00)
yy	The last two digits of the year.	19
yyyy	A four digit year	2019
MM	The number of the month	03

MMM	A short text description of the month	Mar
MMMM	The full name of the month	March
dd	The number of the day	25
ddd	A short text for the day of the week	Mon
dddd	The full name of the day of the week	Monday
/	Date separator. Its appearance depends on regional settings.	

Symbol	Meaning
hh	Hours
mm	Minutes
ss	Seconds
tt	If present represent data in AM/PM format
:	Time separator

- **Custom format Strings for TotalTimeON statistic values**

Symbol	Meaning
hh	Hours
mm	Minutes
ss	Seconds
f	Tenths of Seconds
ff	Hundredths of a second
fff	Milliseconds



You must always use "\" separator character.  
Example:

- dd\hh:mm -> 30.10:20
- hh:mm:ss -> 10:20:52
- mm:ss -> 20:52
- dd\hh:mm:ss\fff -> 30.10:20:52.657



Customized DateTime format settings might not be supported in WebHMI.

### Run Mode

Enabling this will start up the Trend in RUN mode when loaded.

### Linked Pen

This is used to set the name of the linked pen.

### Record Only When Good Quality

When enabled, recording will only be executed in those cases where communication quality is 'Good' when the tag in question is connected to the field by means of a Communication Driver. 0

### Value Label Width

This is used to set the width of the label.

## Data Analysis - Scale

### Scale Padding Factor

This is used to set a padding factor for the space of the margins between the borders and the scales.

### Logarithmic Y Axis Base

This is used to set the base value for the logarithmic scale (use only with previous option enabled).

### Logarithmic Y Axis Scale

This is used to enable the logarithmic scale for the Y Axis.

### Automatic Scale

Enabling this option will activate the scale management at a general level making it unique for all pens by assuming the minimum value from the different pens' minimums and the maximum value from the different pens' maximums. Each pen will then be traced according to the general minimum and maximum value.

## Data Analysis - Aspect

### General Background Color

This is used to set the object's background color.

### Plot Background Color

This is used to set the color of the chart's plot area.

### X/Y Minor Grid Line Color

This is used to select the color of the minor grid lines on the x/y axes.

### Axis Label Color

This is used to change the color of the labels on the XC and Y axes.

#### **Value Label Color**

This is used to set the color of the labels.

#### **X/Y Grid Line Color**

This is used to select the color of the major grid line on the x/y axes.

#### **Pen Thickness**

This is used to set the pen thickness.

#### **Rotated**

This allows the chart area to be display vertically.

#### **Show Date/Time Toolbar**

Enables/disables the date/time button options on Toolbar

#### **Always Expanded**

Enabling this option will expand the data analysis chart area to the whole surface of the window causing the toolbars and other others to go out of view.

#### **Show Statistics Lines**

Enables/Disables the visibility of the statistics lines on the chart (Min, Max, Media).

#### **Measure Unit Multiplier**

This property is used to enlarge the size of the bars and relating measure units by multiplying the actual measure unit with the value set here (default value = 1). Since the measure unit is fixed per second for pen types placed next to each other, if you set the Measure Unit Multiplier to 3 for example, you will get a unit measuring three seconds.



This property has effect on all the chart's pens as only one temporal axis has been set.

When you need to discriminate by pen, you will have to use this management with more than one temporal axes.

#### **Disable zoom behaviour**

This is used to disable the use of the zoom on the chart at runtime.

#### **Legend Area Foreground Color**

This is used to set the legend's text color.

#### **Point Precision**



This property is used to establish the number of decimal figures to display in the tooltip that shows the value of the points when moving the cursor on the curve. This same setting also has effect on values shown on the curve's labels when enabled.

#### **Chart background Color**

This is used to set the color of the chart's background.

#### **Command Timeout**

This is used to indicate the maximum waiting time in seconds to connect the object to the database. Timeout will be 30 second when value is left set at 0.

#### **Show Color Palette Tab**

This is used to enable/disable the toolbar that contains the Palette object.

#### **Max Resolve Overlapping Label Points**

This is used to set the maximum number of labels whose positions will be automatically managed when overlapping.

#### **Max Label Points**

This is used to set the maximum number of label to be displayed in the object.

#### **Auto Hide Toolbar**

This option enables/disables the automatic hide toolbar function.

#### **Collapsed Height**

This is used to set the height of the object when collapsed to toolbar.

#### **Collapsed Width**

This is used to set the width of the object when collapsed to toolbar.

#### **Show Button Toolbar**

Enables/disables the button toolbar.

#### **Show Advanced Settings Toolbar**

Enables/Disables the advance settings/buttons on toolbar.

#### **Show Max Records Toolbar**

Enables/Disables record button options on Toolbar.

#### **Show Compare Toolbar**

Enables/Disables the compare button options on Toolbar.

#### **X Margin for all values**

This is used to set a maximum margin for all values on the X axis.

### Value Label Background

This is used to set the labels' background color.

### User Based Runtime Settings

Enabling this option will personalize the object's style according to the user logged on at runtime.

## Data Analysis - Grid

### X/Y Auto Grid

This option is used to activate the use of the automatic grid scale on the X/Y axes.

### Show X/Y Grid Line

Enabling this will show the major grid lines on the X/Y axes.

### Show X/Y Minor Grid Line

Enabling this will show the minor grid lines on the X/Y axes.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

### **Runtime Configurable**

This displays the toolbars for saving and loading configurations at Runtime.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

### Configurable Write Access Level

This is used to set the minimum access level allowed for editing/configuring the object at runtime.

### Configurable Write Access Mask

This is used to set the access area to allow the object to be edited/configured at runtime.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Runtime Legend Fields

### Visible

This field allows you to set the pen as visible or not visible (pen is visible for default).

### Show Axis

Enabling this field by checking the checkbox will show the pen's customized scale reporting its minimum and maximum values on the right hand side of the chart.

### Statistics

This field allows you to display the statistic curves relating to the pen. In addition to the pen's trend curves, another two curves will also be displayed showing the minimum and maximum value assumed by the variable.

### Color



This indicates the pen's color.

#### **Variable**

This field is used to select the variable to associate to the pen (or the DataLogger column) in cases where the variable is associated to a DataLogger and not a Historian).

#### **Name**

This field is used to enter the name with which the pen will be displayed in the legend. It is also possible to enter a String ID to manage language changes in runtime.

#### **EUnit**

This enables the use of the engineering unit.

#### **Live Value**

This shows the current value.

#### **Nearest Point Date**

#### **Nearest Point Value**

#### **Historian/DataLogger**

This shows the name of the Historian or DataLogger to which the pen's variable has been associated.

#### **Total Data Source**

This shows the maximum number of points that will be loaded in the object's buffer.

#### **Total Compressed Points**

This shows the number of compressed points when the "Enable data aggregation" has been activated.

#### **Compress Ratio**

This indicates the data compression ratio when the "Enable data aggregation" has been activated (for instance, a compression ratio of 10 will show one record for every 10 in the DB in the object).

#### **Min/Max**

These two values define the minimum and maximum pen visualization limits. In cases where the Tag associated to the pen has been associated with an Engineering Unit, the Minimum and Maximum values will be fetched directly from the Engineering Unit and these two fields will be ignored.

#### **Min/Max Statistic**

These two fields show the minimum and maximum statistic values for data loaded in the object's buffer.

#### **Average**

Average value of data loaded in the object's buffer.

#### **Median**

This indicates the value that is lying at the mid point of the total distribution of data loaded in the object's buffer.

Variance

This indicates the value of the variance of the total of data loaded in the object's buffer.

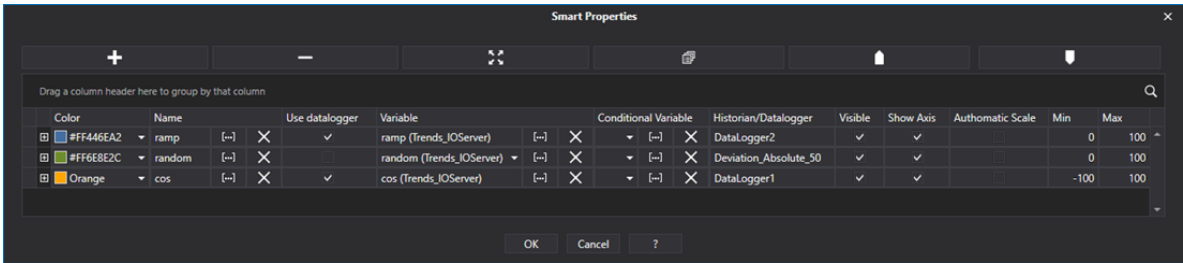
Standard Dev.

This indicated the Standard Deviation on the total of data loaded in the object's buffer.

Add Missing Points

When to True allows the curve to be extended by adding some virtue samples. This is handy when needing to represent a curve associated to a variable sampled upon variation. The use of this property is recommend in combination with the step line pen type.

Smart Properties



- The **Edit** button can be used to access a preview of how the object will be shown during runtime.
- Other pen properties can be accessed by using the button indicated in the above screen.

A pen relating to the curve to be plotted can be added by clicking the "+" button and set by using the proposed columns from left to right:

Color

This is used to select a color for the pen.

Name

This field is used to insert the name with which the pen will be displayed in the legend area. It is also possible to insert a String ID to manage language changes at runtime.

Use DataLogger

When this checkbox is disabled, it will be possible to select the pen's Tag from those which have been associated to a Historian. When this checkbox has been enabled, it will be possible to select the Tag from those associated to a DataLogger. In the latter case, a DataLogger column will be selected.

Tag

This field is used to select the variable to associated to the pen (or Datalogger column when the variable has been associated to a DataLogger and not a Historian).

### Conditional Tag

This field is used to select a String type Tag whose text will be used as a filter to extract data of the pen linked to the Datalogger (this does not function with pens linked to Historians). The filter is done by adding a WHERE clause to the read query with the content of the conditional tag.

For example, if needing to designate the trend of a conditional Tag to a production batch, it would be possible to set the Tag with string: [Column\_Lotto] = '202105-T2'.

### Historian/DataLogger

This shows the name of the Historian or Datalogger to which the pen's variable has been associated.

### Visible

This checkbox is checked for default to make the pen visible. Unchecking this box will make it invisible.

### Show Axis

Enabling this field by checking the checkbox will display the pen's customized scale, which shows the pen's minimum and maximum values, on the right of the chart.

### Automatic Scale

Selecting this checkbox will display the pen's scale, if enabled, automatically and not with the fixed Minimum and Maximum values.

### Min, Max

These two values define the pen's minimum and maximum visualization limits. In cases in which the Tag associated to the pen has been associated with an Engineering Unit, the Minimum and Maximum values will be fetched directly from the Engineering Unit and these two fields will be ignored.

### Use Source TimeStamp

When enabling this checkbox the pen chart will be plotted using the variable's colonna SourceTimeStamp column and not the record's RecordDateTime column. You must enable the Source TimeStamp column for the Tag in those cases where the pen is connected to a DataLogger. The SourceTimeStamp column is recorded with UTC time for both the Historian and DataLoggers.

### Use Local TimeStamp

This box can only be enabled when the "Use Source TimeStamp" box has been checked. When both boxes are checked, the pen's chart will be plotted using the SourceTimeStamp column applying local time as well.

### Use Aggregated Tables

When the datalogger's "Use aggregated tables" property has been activated, it will be possible to display the aggregated tables as the chart's data source.

### Show Min Pen

This is used to show the curve given by the aggregated table's minimum value.

### Show Avg Pen

This is used to show the curve given by the aggregated table's average value.

### Show Max Pen

This is used to show the curve given by the aggregated table's maximum value.

### Statistics

Enabling this field by checking the checkbox will make the statistics curve relating to the pen visible. In addition to the pen's trend curve, another two curves will be displayed showing the maximum and minimum values obtained by the variable.

### Log Scale

Select this to enable the logarithmic scale.

### Log Base

This is used to set the logarithmic base to apply to the logarithmic scale when selected.

### Use EU Min/Max

When enabling this field by checking the checkbox, the variable's associated EU min. and max. values will be used instead of the min. and max. values it was set with.

### Add Missing Points

Enabling this field by checking the checkbox will extend the represented curve by adding the missing samples. This is handy when needing to represent a curve associated to a variable sampled upon variation. The use of this property is recommended in combination with the Step Line pen type.

### Type

This field is used to select the style with which to display the pen.

### Array Index

This field is used to set the array index when variable is array type.

### Line Thickness

This field is used to set the thickness of the pen's line.

## OPC UA Data Trend

This object is used exclusively for representing historical data (in pause mode) and enables users to set filters for displaying data within the desired time ranges.

It is therefore essential that the tag associated to the object be linked to a historical resource. This object receives data directly from the Server and does not have direct access to the Data Base unlike the Data Analysis and Real Time Trends objects.

This object provides the user with the Fetch, Next and Stop buttons when functioning in runtime mode. These buttons allow the user to browse the data ranges displayed within the graph. The amount of data to be displayed can be set in the 'Enable Max Return Values' property found in the toolbar Option on the left. Take care when doing this as the oldest data is always loaded according to the defined time range during the data loading phase. When the number of data samples for the time range exceeds the set limit you can use the Next buttons to scroll back and forth. For example, if you set a limit of 25 samples and a time range = 1 hour, but a 100 samples are recorded within the hour instead, the Trend will display the first 25. However, when you use the Next key, the next 25 samples will be loaded and so forth until the last available recorded samples have been reached.



As a difference to the RealTime Trend, this object also has an 'Options' toolbar that can be used for setting data selections and filters.



The Historical Trend object can only display one pen only which can also be configured by dragging and dropping the desired variable in the object.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## OPC Trend - Settings

### Layout Configuration

This is used to configure the layout with which the object will be presented by using the Options window. Within this window you will find the following properties:

- **Enable Start Time:** This is used to set the start date and time.
- **Enable End Time:** This is used to set the sampling end date and time.
- **Enable Max Return Values:** This is used to set the max. number of read values.
- **Read Type:** This is used to set read type.
- **Processed Aggregation Type:** This is used to specify which mode to use for aggregating data from the database when not all the data can be displayed in the trend.

### View Mode

This is used to select which mode to view data (chart, grid).

### Title

This is used to set a title for the object.

### Filter Type

This is used to set the filter type by period with which to load data.

#### **Max. History Count**

This is used to set the maximum number of recorded samples.

#### **Read Type Mode**

This is used to select which read type mode to use.

#### **Start Time**

This is used to set a start time/date.

#### **End Time**

This is used to set a end time/date.

#### **Use Start Time**

When enabling this, the settings for the start time/date will be used.

#### **Use End Time**

When enabling this, the settings for the end time/date will be used.

#### **Use Max Return Values**

This is used to limit the number of loaded historical values.

#### **Max Return Values**

This is used to set the maximum number of values to request server for.

#### **Decimal Number Cursor**

This is used to specify the number of decimals present in the cursor's value.

#### **Fetch Data On Screen Load**

This forces the screen to display only when the data has been completely loaded in the object.

## **OPC Trend - Scale**

#### **Automatic Scale**

This is used to enable the automatic scale of the chart's pens.

#### **Min. Scale Value**

This is used to set the minimum value for the pen scale.

#### **Max. Scale Value**

This is used to set the maximum value for the pen scale.

#### **Y-Axis Prefix Text**

This is used to set insert a specific field before the value of the axis (letter, sign, formulae, etc.).

#### Show Y Major Grid line

Enabling this will show the major grid line on the Y axis.

#### Show Y Minor Grid line

Enabling this will show the minor grid line on the Y axis.

#### Show X Major Grid line

Enabling this will show the major grid line on the X axis.

#### Show X Minor Grid line

Enabling this will show the minor grid line on the X axis.

### OPC Trend - Aspect

#### Plot Background Color

This is used to set the color of the Plot area's background.

#### Trend Title Font Color

This is used to set the title's text font color.

#### Button Font Color

This is used to set the color of the button's text font color.

#### Touch Screen System

This enables the object's interactivity with touchscreen systems.

#### Graphic Buttons

This replaces the text in buttons with intuitive icons.

#### Show Labels

Enabling this will show labels on the chart for each curve plot point.

#### Show Plot Markers

Enabling this will show a marker for each curve plot point.

#### Show Runtime Tool Bar

This enables the runtime tool bar to show.

#### Auto-Hide Toolbar

Select this to auto-hide the toolbar when mouse cursor moves away from its vicinity.

#### Always Expanded

Selecting this will always keep the object expanded to its maximized size hiding the other areas such as the toolbar which will remain invisible.

### **Plot Marker Size**

This is used to set the size of the plot markers on the chart.

### **Plot Transparency**

This is used to set the transparency of the curve area on the chart.

### **Collapsed Height**

This is used to set the maximum height of the object when collapsed from the Toolbar.

### **Collapsed Width**

This is used to set the maximum width of the object when collapsed from the Toolbar.

### **Show Crosshair lines**

When enabled, crosshair lines will show on the targeted sampled value (chart mode).

### **Show Crosshair Argument Labels**

Selecting this will enable the argument labels with values to show on the axes.

### **Show Crosshair Value Label**

This refers to crosshair lines and when selected shows the value label of the targeted sampled value.

### **Max Label Number**

This property allows you to set the maximum number of plot points for which labels can be displayed for. For example, if you enable label to show but load a number of samples that exceed the maximum number set here, the labels will automatically be disabled. Please bear in mind that a high number of displayed labels will effect the performance of the graphical interface by making it slower. It is therefore advised not to set this property higher than the one set for default otherwise the interface will become very slow to perform or freeze while loading data.

## **Style**

### **Background Color**

This is used to set the container's background color.

### **Text Color**

This is used to set the object's text color.

### **Horizontal Content Alignment**

This is used to horizontally align all the contents within the object.

### **Vertical Content Alignment**

This is used to vertically align all the contents within the object.



### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

### **Trend Title Font Style**

This is used to set the style of the title's font.

### **Button Font Settings**

This is used to set the object's main font.

### **Y-Axis Font Settings**

This is used to set the font type to use for the X axis (chart mode).

### **X-Axis Font Settings**

This is used to set the font type to use for the Y axis (chart mode).

## **Layout**

### **Position X**

This is used to set the object's horizontal position in pixels.

### **Position Y**

This is used to set the object's vertical position in pixels.

### **Height**

This is used to change the object's height.

### **Width**

This is used to change the object's width.

### **Border Thickness**

This is used to set the container's border thickness.

### **Margin**

This is used to set the object's margins.

### **Lock Object Position**

This locks the object's position on screen.

## **Visibility**

### **Transparency**

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### Force as Dynamic on Client

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

#### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

#### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

#### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

#### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

#### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

#### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

#### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

#### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

#### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

#### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## User Access

### Access Level

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### User Access Role

This is used to set the name of the user password group to allow user interaction with this object.

### Access Level from Tag

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Sparkline Chart

The "Sparkline Chart" object is a graphical chart that displays the values of the current day only. The DB table from which the data is retrieved must have at least one value column (see "value column name" property), and a date/time column (see "time column name" property)

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Sparkline - Settings

### Data Source

This is used to indicate the source from which data will be fetched; DB connection string.

The following properties can also be set using the "**Control DataSource Editor**" dialog window:

- **Select:** this is used to select the table from which to fetch data.
- **Max. Take (Nr. Records):** this is used to set the maximum number of records to be displayed (96 for default which is the number of a quarter hours in a day). This can be activated from the table selection window.
- **Where:** filter options for selecting other data sources from where to fetch data to display in cases where multiple data sources are being used.
- **Group By:** This corresponds to the query's group by clause.
- **Sort:** This is used to sort data based on the selected reference column.
- **Data Column List:** name of the column containing the valued to be reproduced on the chart.
- **Time Column List:** time column column name.

### Reload Data Frequency

This is used to at what frequency data is to be refreshed.

### Label Format

This is used to set the label format.

### Invalid Data Value

This is used to select the value to use for invalid data with floating point.

## Sparkline - Scale

**Min. Value**

This is used to set the minimum value on the Y axis.

**Max. Value**

This is used to set the maximum value on the Y axis.

**X Axis Title**

This is used to set the title of the X axis.

**Y Axis Title**

This is used to set the title of the Y Axis.

**X Minor Div. Number**

This is used to set the number of minor dividers on the X axis.

**X Major Div. Number**

This is used to set the number of major dividers on the X axis.

**Y Minor Div. Number**

This is used to set the number of minor dividers on the Y axis.

**Y Major Div. Number**

This is used to set the number of major dividers on the Y axis.

**Automatic Scale**

This is used to set the automatic scale for the Y axis.

## **Sparkline - Aspect**

**Plot Border Color**

This is used to set the color of the chart's border.

**Label Color**

This is used to set the color of the labels.

**Pen Color**

This is used to set the color of the pens.

**Marker Color**

This set the color of the markers.

**Grid Color**

This is used to set the color of the grid in the chart area.

**Show Markers**

Enables markers to show.



### Show Labels

Enables labels to show.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### **Text Direction**

This allows you to set the direction of the text within the object.

### **Enable UI**

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

#### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

#### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

#### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

#### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

#### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## **Animations**

#### **Animations**

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## **Commands**

#### **Commands**

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## State Bar Chart

The State Bar Chart object is used for representing states (values) of variables associated to a DataLogger or an Historian in order to graphically view, for example, the operational state of a machine.

You can also define threshold colors which the pen will assume according to the variable value associated to a specific time. If the set base times have multiple values, the dominating value will be represented as a back color while the one immediately after (see image below), will be represented in the border color for duration (clicking on the cell which represents inhomogeneous values, the time interval for the relative values will display).

However, you can define as many pens as desired to represent several states at the same time. The chart can also be scrolled to show the historical part as well. .



## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Chart - Settings

### Filter Type

This is used to set the time filter type to display data.

### No. Samples

This is used to set the maximum number of samples to be acquired from the DB the moment the pen is represented.

### DB Connection String

This is used to set the DB connection string.

### Object Configuration

See chapter below on Smart Properties

### Layout Configuration

This is used to configure the 'Time Range' and 'Legend' windows for the object when in runtime mode.

## Chart - Aspect

### Pen Height

This is used to set the pens' height in pixels.

### Separator Height

This is used to set the separator height in pixels.

### Label Color

This is used to select a color for the labels and time lines on the chart.

### Legend Text Color

This is used to select Legend's text color.

### Auto refresh

Select this to refresh values automatically.

### Auto Hide Toolbar

This is used to enable the auto hide toolbar function.

### Show Legend Area

This is used to Show/Hide the Legend.



### **Use Absolute Ranges**

Permits data to be shown between the relative/absolute ranges.

### **Show Row Labels**

This is used to show/hide the labels.

### **Show Tooltips**

Allows tooltips for data displayed in the object to be hidden when hovering mouse over the pen. Therefore, the value assumed by the variable at that moment will not be shown.

### **Command Timeout**

This is used to set a DB Connection timeout.

### **User Based Runtime Settings**

Enabling this option will customize the object's style at runtime based on the user.

## **Style**

### **Background Color**

This is used to set the container's background color.

### **Text Color**

This is used to set the object's text color.

### **Horizontal Content Alignment**

This is used to horizontally align all the contents within the object.

### **Vertical Content Alignment**

This is used to vertically align all the contents within the object.

### **Disable Anti-Aliasing**

This is used to Enable/Disable the Anti-aliasing function.

## **Fonts**

### **Font Size**

This is used to set the size of the selected font.

### **Font Family**

This is used to set a font type to use for text.

### **Font Style**

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### **Menu Activation**

This allows you to set the name of the contextual menu that shows when right clicking the object.

### **Is Interactive**

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### **Menu with Left button**

This enables the contextual menu to show by also using the left mouse key.

### **TAB Stop**

This makes the object selectable at runtime using the TAB button.

### **Text Direction**

This allows you to set the direction of the text within the object.

### **Enable UI**

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

### Preserve Size

This is used to preserve the values of the local size property group when the object is updated from the library.

### Preserve Commands

This is used to preserve the list of commands when the object is updated from the library.

### Preserve Animations

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Smart Properties

### Name

The name to display the pen in the legend with is entered here. It is also possible to enter String ID to manage language changes at runtime.

### Variable

This field is used for selecting a variable to associate to the pen (or DataLogger column when the variable is associated to a DataLogger and not a Historian).

### Use DataLogger

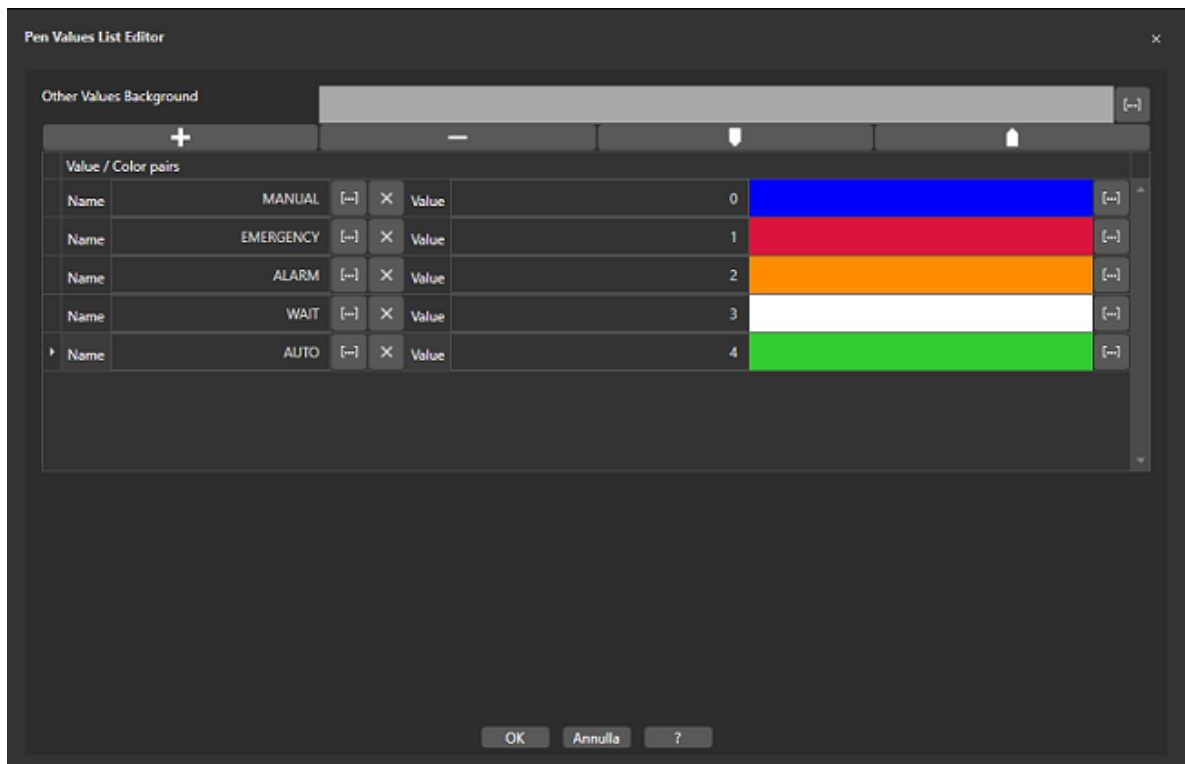
When this checkbox is left unmarked, it will make it possible to select the pen's tag from those associated with a Historian. When this checkbox is marked, it will be possible to select the pen's Tag from those associated with a Datalogger. In the latter case, a DataLogger column will be selected.

### DataLogger/Historian

This shows the name of the Datalogger to which the pen's variable is associated.

### Value/Color Pairs

This is used to set the color that will be assigned to the pen based on the value. In cases where the value obtained by the pen has not been defined, the color from the "other value background" option will be used instead (see screen shot below).



## RealTimeTrend

As a difference to the other objects of this type, this object displays the variables linked to the chart's pens in real-time. Therefore, pens that relate to variables not necessarily connected to a historical resource can be configured.



When in pause mode, the realtime trend will stop displaying the values in realtime in order to perform an analysis on the currently displayed data.



When putting the Trend object in stop mode, pens can be selected from the legend by clicking on the corresponding line of the pen desired. This will highlight the corresponding curve on the chart. To deselect the pen, simply click on the pen's line again while pressing down the CTRL key.

The Realtime Trend object provides various navigation buttons for the user to use in runtime to display data. The user can expand the range of displayed data on screen by using the scroll bar on the X and Y axis (eg. the time interval presented on the Trend's X axis can be modified).

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions



This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Trend - Settings

### Object Configuration

Please see the below chapter on Smart Properties

### Layout Configuration

This is used to configure the 'Grid' and Legend' windows for the object in runtime mode.

### Runtime Mode

This is used to set runtime function type.

### Record Every

This is used to set the when to record to update the curve in the chart.

### Show Time Frame

This is used to set a time frame to display in the chart.

### Num. Samples

This is used to set the maximum number of samples.

### Pen Name

This is used to set the name of the pen.

### Point Precision

This property is sued to establish the number of decimal figures to be displayed in the tooltip that shows the value of points when stopping the Trend and moving the cursor along the curve.

### Date Format

This is used to set the Date Format.

### Time Format

This is used to set the Time Format.

- **Standard format strings for Date/Time values**

The standard Date/Time format strings contain one character. This character defines the model used for representing the value (and in which mode to display the number of years, months and so on).

<b>Format Specifier</b>	<b>Description</b>	<b>Result of formatting a sample value (March 25 2019 12:00:00)</b>
d	Short date pattern	3/25/2019
D	Long date pattern	Monday, March 25, 2019
t	Short time pattern	12:00 AM
T	Long time pattern	12:00:00 AM
f	Full date/time pattern	Monday, March 25, 2003 12:00 AM
F	Full date/time pattern	Monday, March 25, 2003 12:00:00 AM
g	General date/time pattern	3/25/2019 12:00 AM
G	General date/time pattern	3/25/2019 12:00:00 AM

- **Custom format Strings for Date/Time**

To create models for Date/Time values, you will need to combine the strings listed in the table below. These strings represent the year, month, day, number and so on in different formats.

<b>Symbol</b>	<b>Meaning</b>	<b>Result of formatting a sample value (March 25 2019 12:00:00)</b>
yy	The last two digits of the year.	19
yyyy	A four digit year	2019
MM	The number of the month	03
MMM	A short text description of the month	Mar
MMMM	The full name of the month	March
dd	The number of the day	25
ddd	A short text for the day of the week	Mon

dddd	The full name of the day of the week	Monday
/	Date separator. Its appearance depends on regional settings.	

Symbol	Meaning
hh	Hours
mm	Minutes
ss	Seconds
tt	If present represent data in AM/PM format
:	Time separator

- **Custom format Strings for TotalTimeON statistic values**

Symbol	Meaning
hh	Hours
mm	Minutes
ss	Seconds
f	Tenths of Seconds
ff	Hundredths of a second
fff	Milliseconds



You must always use "\" separator character.

Example:

- dd\hh:mm -> 30.10:20
- hh:mm:ss -> 10:20:52
- mm:ss -> 20:52
- dd\hh:mm:ss\fff -> 30.10:20:52.657



Customized DateTime format settings might not be supported in WebHMI.

### Record Only When Good Quality

This is used to enable/disable the recording of the tag only when tag has 'good' quality.

### User Based Runtime Settings

Enabling this option will customize the object's style at runtime based on the user.

## XY Trend - Scale

### Axis Line Color

This is used to set the color of the lines of the Trend's axes.

### Grid Color

This is used to set the color of the grid.

### Plot Border Color

This is used to set the color of the chart's plot area borderline.

### Axis Label Text Color

This is used to set the label text color on the chart's axes.

### Axis -stroke Thickness

This is used to set the thickness of the Trend's axes' lines.

### Number of Major X Axis ticks

This is used to set a number of major dividers on the Plot area's X axis. If the 'Automatic Scale' property has been enabled, the value set will be considered as the number of intervals to keep to. However, it is not guaranteed that this value is respected in rescaling operations.

### Number of Major Y Axis ticks

This is used to set the number of major divider lines on the Plot area's Y axis. If the 'Automatic Scale' property has been enabled, the value set will be considered as the number of intervals to keep to. However, it is not guaranteed that this value is respected in rescaling operations.

### Number of Minor Y Axis ticks

This is used to set the number of minor grid lines on the Y axis.

### Automatic Scale

When enabling this, the automatic scale will be used for the pens on the chart.

### Show X Major Grid Lines

Enabling this will show the major grid lines on the X axis.

### Show Y Major Grid Lines

Enabling this will show the major grid lines on the Y axis.

### Plot Border Thickness

This is used to set the thickness of the chart's plot area borderline.

### Scale Padding Factor

This is used to set the margin space between the borders and the scales.

## Trend - Aspect

### Plot Area Background Color

This is used to select a color for the Plot area's background.

### Trend Background Color

This is used to select a background color for the trend.

### Pen Area Background Color

This is used to select a color for the pen area's background.

### Legend Text Color

This is used to change the color of the legend's text.

### Trend Orientation

This is used to set the trend's orientation type (horizontal or vertical)

### Show Pen Name

Enabling this will show the pen's name on the label showing underneath the cursor.

### Show Pen Area

This is used to show/hide the pen area within the object.

### Show Legend

This is used to show/hide the object's legend area.

### Auto Hide Toolbar

This enables the toolbar to automatically hide.

### Always Expanded

When enabled, the object will display always expanded to maximum size.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

### **Y-Axis Font**

This is used to set the font to use for the Y-axis (Chart mode).

### **X-Axis Font**

This is used to set the font to use for the X-axis (Chart mode).

### **Value Label Font**

This is used to set the text font of the value labels shown on the chart.

## **Layout**

### **Position X**

This is used to set the object's horizontal position in pixels.

### **Position Y**

This is used to set the object's vertical position in pixels.

### **Height**

This is used to change the object's height.

### **Width**

This is used to change the object's width.

### **Border Thickness**

This is used to set the container's border thickness.

### **Margin**

This is used to set the object's margins.

### **Lock Object Position**

This locks the object's position on screen.

## **Visibility**

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.



### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### Preserve Text

This is used to preserve the added text string when the object is updated from the library.

## User Access

### Access Level

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### User Access Role

This is used to set the name of the user password group to allow user interaction with this object.

### Access Level from Tag

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Smart Properties

### Visible

This checkbox is marked for default to enable the pen's visibility. When unmarked, the pen will be made invisible.

### Pen Color

This is used to indicated the pen's color.

### Pen Name

This field is used to enter the name to display the pen with in the legend. A String ID can also be entered to manage language changes at runtime.

### Tag Name

This field is used for selecting the variable to associated to the pen (or the DataLogger column in cases when the variable is associated to a DataLogger and not a Historian).

### Show Axis

Enabling this field by checking the checkbox will display the pen's customized scale, which shows the pen's minimum and maximum values, on the right of the chart.

### Automatic Scale

Selecting this checkbox will display the pen's scale, if enabled, automatically and not with the fixed Minimum and Maximum values.

### Min/Max

These two values define the minimum and maximum limits for displaying the pen with. In case where the Tag associated to the pen has been associated with an Engineering Unit, the minimum and maximum values will be taken directly from the Engineering Unit and these two field will be ignored.

### Use EU Range

This enables the use of the engineering range.

### Array Index

This field is used to set the array index when variable is array type.

### Thickness

This field is used to set the thickness of the pen's line.

### Pen Style

This field is used to select the style to show the pens with.

## Runtime Legend Fields

### Visible

This checkbox is used to enable the pen's visibility (visible for default).

### Pen Color

This indicated the pen's color.

### Pen Name

The name with which the pen will be displayed in the legend is entered here. You can also enter a String ID to manage language changes at runtime.

### EUnit

This is used to enable the use of the engineering unit.

### Date

### Value

This is the realtime value assumed by the variable associated to the pen.

### Min/Max

These two values define the minimum and maximum limits for displaying the pen with. In case where the Tag associated to the pen has been associated with an Engineering Unit, the minimum and maximum values will be taken directly from the Engineering Unit and these two field will be ignored.

### Min/Max Statistic

These fields indicated the minimum and maximum statistics values for data loaded in the object's buffer.

### Average

Average value of data loaded in the object's buffer.

## 2.29. Viewers

### OPC Events Viewer

The "OPC Events Viewer" window is an object used for displaying events that have been notified by the connected OPC server.

The reported event types are tightly linked to the Server's programming.



For example, the AUDIT window may also contain notifications as a consequence of alarm ACK or RESET events.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## OPC Events Viewer

### Allow Columns to be Dropped

Enabling this will allow columns to be removed at Runtime.

### Layout Configuration

This is used to change the object's layout by adding/removing columns to be viewed at runtime.

### User Based Runtime Settings

enabling this will allow the object to be customized based to user logged on at runtime.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more languages defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### **Position Y**

This is used to set the object's vertical position in pixels.

### **Height**

This is used to change the object's height.

### **Width**

This is used to change the object's width.

### **Border Thickness**

This is used to set the container's border thickness.

### **Margin**

This is used to set the object's margins.

### **Lock Object Position**

This locks the object's position on screen.

## **Visibility**

### **Transparency**

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

### **Visibility**

This is used to enable the object's visibility.

### **Visible on Web Client**

This makes the object visible on Web Client.

### **Visibility Layer Level**

This is used to set the mask that determines the object's visibility.

### **Zoom Level Visibility X**

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### **Zoom Level Visibility Y**

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## **General**

This is a list of the general properties which are the same for objects and draws:



### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### Force as Dynamic on Client

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

### Preserve Size

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

### **Runtime Configurable**

Selecting this will allow the object to be modified at Runtime. This option also allows the configuration of the created object to be saved and called.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

### Configurable Write Access Level

This is used to set the minimum access level allowed for editing/configuring the object at runtime.

### Configurable Write Access Mask

This is used to set the access area to allow the object to be edited/configured at runtime.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Calculator

The Movicon NExT calculator object has all the functions of a scientific calculator and, like all the other viewer objects, can be inserted on screen and configured through its various properties.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more languages defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### **TAB Stop**

This makes the object selectable at runtime using the TAB button.

### **Text Direction**

This allows you to set the direction of the text within the object.

### **Enable UI**

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**



This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project.

This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## **Animations**

### **Animations**

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Cross Reference Viewer

For further information on this object please see the topic relating to "Cross\_Reference"

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### **Disable Anti-Aliasing**

This is used to Enable/Disable the Anti-aliasing function.

## **Layout**

### **Position X**

This is used to set the object's horizontal position in pixels.

### **Position Y**

This is used to set the object's vertical position in pixels.

### **Height**

This is used to change the object's height.

### **Width**

This is used to change the object's width.

### **Border Thickness**

This is used to set the container's border thickness.

### **Margin**

This is used to set the object's margins.

### **Lock Object Position**

This locks the object's position on screen.

## **Visibility**

### **Transparency**

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

### **Visibility**

This is used to enable the object's visibility.

### **Visible on Web Client**

This makes the object visible on Web Client.

### **Visibility Layer Level**

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### Force as Dynamic on Client

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

#### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

#### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

#### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

#### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

#### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

#### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

#### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

#### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

#### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### User Access Role

This is used to set the name of the user password group to allow user interaction with this object.

### Access Level from Tag

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## OPC Client Status Viewer

This object is found in the Viewers Miscellaneous category and can be invoked when starting up the project in runtime from development mode with the "Shift + F1" combo keys. .

This window shows the connected OPC Client status.

Those clients that are connected to the local machine are highlighted in the window started up on the client.



This image shows the "MoviconNExT – Browser..." which is the opc browser window opened from development mode, while the "MoviconNExT – asOpcStatus – asOpcStatusClient\_IOServer" is the project's actual screen.

The sessions are the connections made by local clients, in this case there are two sessions in progress indicated by the yellow triangle.

The number of subscriptions refers to the subscribed group information and are highlighted with the red circle.

The monitored items are the subnodes relating to the variables put into use with additional subscribed information such as "SecondsTillShutdown" that indicates the remaining seconds before a redundant server or 'Server' shuts down.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NEXT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### **Position Y**

This is used to set the object's vertical position in pixels.

### **Height**

This is used to change the object's height.

### **Width**

This is used to change the object's width.

### **Border Thickness**

This is used to set the container's border thickness.

### **Margin**

This is used to set the object's margins.

### **Lock Object Position**

This locks the object's position on screen.

## **Visibility**

### **Transparency**

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

### **Visibility**

This is used to enable the object's visibility.

### **Visible on Web Client**

This makes the object visible on Web Client.

### **Visibility Layer Level**

This is used to set the mask that determines the object's visibility.

### **Zoom Level Visibility X**

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### **Zoom Level Visibility Y**

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## **General**

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### Force as Dynamic on Client

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

### Preserve Size

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## PDF Viewer

The "PDF Viewer" enables associated PDF documents to be viewed in runtime. Note that in this case the pages will be bigger than the screen window layout and, therefore, side scrollbars will be provided to make navigating the various pages easier for the user.



The "PDF Viewer" management is not supported during Web Visualization. In cases where the object is deployed on the web client, the "Save", "Open", "Print" and "Close" functions will not function and the corresponding buttons will remain disabled.

### Limitations on WebHMI Client

The "Open" and "Close" buttons are not supported in WebHMI.

The "Save with Name", "Print" and "Find" options are provided by the predefined client/mobile platform browser's PDF viewer associated to the PDF files: when clicking on these buttons, the PDF will be opened/downloaded in a separate card/window within a specific viewer application according to the browser's settings.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## PDF Viewer Settings

### PDF source file

This is used to set the path of the PDF file which will open at runtime. In the WebHMI Client the source file should be located in the target machine. We suggest you copy the PDF source file directly in the project resources folder. In this way, the PDF document will be transferred with the project when loaded in the target machine by the Deploy Server .

### Command Bar Style

This is used to set the style of the command bar.

### Page Number

This is used to set the viewer to open showing a certain page.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.



## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more languages defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed.

## Layout

**Position X**

This is used to set the object's horizontal position in pixels.

**Position Y**

This is used to set the object's vertical position in pixels.

**Height**

This is used to change the object's height.

**Width**

This is used to change the object's width.

**Border Thickness**

This is used to set the container's border thickness.

**Margin**

This is used to set the object's margins.

**Lock Object Position**

This locks the object's position on screen.

## Visibility

**Transparency**

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

**Visibility**

This is used to enable the object's visibility.

**Visible on Web Client**

This makes the object visible on Web Client.

**Visibility Layer Level**

This is used to set the mask that determines the object's visibility.

**Zoom Level Visibility X**

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

**Zoom Level Visibility Y**

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a

value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### Force as Dynamic on Client

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors

This is used to preserve the values of the local color property group when the object is updated from the library.

### Preserve Size

This is used to preserve the values of the local size property group when the object is updated from the library.

#### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

#### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

#### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

#### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

#### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

#### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

#### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

#### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

#### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Report Viewer

The 'Report Viewer' is used for displaying a selected report layout in a specific part of the screen. Scrollbars will be enabled to facilitate navigation when the report layout size is bigger than the Viewer object.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Report Settings

### Report Name

This is used to set the name of the report to link to the viewer.

### Parameters

This is used to set the report's parameters.

### Parameter Loading Timeout

The timeout time for loading report parameters is entered here. Once the timeout has expired, a message will show stating that the report cannot be launched because values of its parameters did not load before timeout expired.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more languages defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NEXT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height



This is used to change the object's height.

### **Width**

This is used to change the object's width.

### **Border Thickness**

This is used to set the container's border thickness.

### **Margin**

This is used to set the object's margins.

### **Lock Object Position**

This locks the object's position on screen.

## **Visibility**

### **Transparency**

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

### **Visibility**

This is used to enable the object's visibility.

### **Visible on Web Client**

This makes the object visible on Web Client.

### **Visibility Layer Level**

This is used to set the mask that determines the object's visibility.

### **Zoom Level Visibility X**

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### **Zoom Level Visibility Y**

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## **General**

This is a list of the general properties which are the same for objects and draws:

### **ToolTip**

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Spreadsheet Viewer

The Spreadsheet Viewer object is used for displaying or modifying associated excel spreadsheets. If the spreadsheet is larger than the viewer screen, side scroll bars will display to facilitate user navigation.



There are some restrictions as regards to using the object in web client. When clicking on the control, it will navigate through several spreadsheets. This navigation is cyclical. Therefore, when the last spreadsheet is reached, it will return back to the first one.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Spreadsheet Settings

### Object Configuration

This property gives access to a window that will allow you to connect variables to cells and spreadsheets.

### Layout Configuration

This is used to modify the object's layout at runtime like an Excel editor.

### Document Path

This is used to set the path from which to load the document.

### Read Only

This is used to set the displayed document to read only in order to stop users from modifying its contents.

### Show Ribbons

Enabling this option will show the ribbons within the window to edit the spreadsheet page.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### **Height**

This is used to change the object's height.

### **Width**

This is used to change the object's width.

### **Border Thickness**

This is used to set the container's border thickness.

### **Margin**

This is used to set the object's margins.

### **Lock Object Position**

This locks the object's position on screen.

## **Visibility**

### **Transparency**

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

### **Visibility**

This is used to enable the object's visibility.

### **Visible on Web Client**

This makes the object visible on Web Client.

### **Visibility Layer Level**

This is used to set the mask that determines the object's visibility.

### **Zoom Level Visibility X**

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### **Zoom Level Visibility Y**

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## **General**

This is a list of the general properties which are the same for objects and draws:

### **ToolTip**

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.





For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Web Browser Viewer

The "Web Browser Viewer" is used to display or modify associated web pages in runtime. If the web page is larger than the screen layout, side scroll bars will appear to facilitate navigation for the user.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## WEB Connection Address

### Web Address

This is used to set the URL address to open within the viewer.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family

- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NEXt 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

#### **Enable UI**

This is enabled for default to allow user interactivity with the object in the UI.

#### **Is Focusable**

Activates / Deactivates focus on the object.

#### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

#### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

#### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

#### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

#### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

#### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

#### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

#### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

#### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:



- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

#### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

#### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

#### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

#### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

#### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## **Animations**

#### **Animations**

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## **Commands**

#### **Commands**

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## 2.30. Historical Data Viewers

### Audit Trail Validator

This object allows you to select a protected data source and choose within which time range to validate data (see the '**Enable Data File Protection**' of the historian or data logger).

The object will display data in table format along with the relevant validation status watermark (see screen shot above). Any manipulated data will be highlighted with a background color, which can be selected from the object's properties (red is used for default), with the corresponding watermark signalling that data has been manipulated.

The object also offers the possibility to print data using the button with the printer symbol on the top right.

- **Start History validation:** starts analyzing data for validation.
- **Abort History Validation:** interrupts data analysis.
- **Abort when First Row is Invalid:** terminates analysis upon detecting an error by reporting the first verified anomaly.



When opting to use the data protection, by enabling the Enable Data File Protection property, we recommend that you set a very high Max Audit Age of at least 10000 days to avoid cyclic deletion of older data.

### Tag

#### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

#### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

# Audit Trail Validator

## Runtime Configurable

Enabling this will allow the object to be configured at runtime using a specific modal window.

## User Based Runtime Settings

Enabling this option will allow the object's style to be customized according to the user logged in at runtime.

## Max. Errors before Aborting

This property is used to set the maximum number of errors before aborting operation as failed data validation.

## Execute Query Timeout

This is used to set a Timeout for validating data.

## Allow Backup File Selection

This is used to enable the selection of backup files at runtime.

## Export Data

This is used to enable data exporting.

## Add Backup File Command

This allows a backup file to added at runtime of the database wished to be validated.

## Layout Configuration

This allows you to configure the object's layout.

## Validation Source

This is used to set the source where the data to be validated resides (Data logger, Historian or Audit).

## Load Filter Type

This is used to select the filter type to apply to the object upon loading.

## Read Data on loading

Enabling this will allow data to be ready when loading object.

## Graphic Buttons

Enabling this will use icons instead of text for the buttons on the bottom bar.



The 'Description (optional)' column can be added to the object's layout to display the Tag's description and therefore allow easier identification of the tag subjected to editing. This column is not part of the table under validation.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more languages defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight



In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### Preserve Text

This is used to preserve the added text string when the object is updated from the library.

## User Access

### Access Level

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### User Access Role

This is used to set the name of the user password group to allow user interaction with this object.

### Access Level from Tag

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Data Logger Viewer

The Movicon.NExT Data Logger Viewer is a configurable object designed for displaying data recorded by the project's Data Loggers.

The Data Logger Viewer is available from the Toolbox 'Viewers' Group.

Any desired number of Data Logger Viewer objects can be inserted in various screens within the project as needed.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## DataLogger Viewer Settings

### Layout Configuration

This allows you to configure the object's layout.

### DataLogger Name

This is used to set the name of the datalogger to view.

### DB Connection String

This is used to set the string for the referenced DB connection.

### Filter Type

Set a time filter for loading data from the database.

### Max Rows

This is used to set the maximum number of rows to show in the object.

### Automatic Update Time

This is used to set the automatic data update time. Leaving this property set with the value = 0, will disable the automatic updating of data. Update time is expressed in seconds.



**Warning!** this feature is only supported in WebHMI Client.

#### Show Best Fit Button

Enabling this option will show the Best Fit Button on the Grid object's top bar and which is used to automatically adapt columns within object.

#### Show Group Box

Enabling this option will show the Group Box.

#### Show Search Box

Enabling this option will show the Search Box.

#### Show Filter Box

Enabling this will show the Filter Box.

#### Show Command Buttons

Enabling this option will show the command buttons.

#### Command Timeout

This is used to set the maximum time to wait for the object to connect to the database in seconds. When left set at 0, timeout will be 30 seconds.

#### User Based Runtime Settings

Enabling this option will allow the object's style to be customized according to the user logged in at runtime.

#### Show Best Fit Button

## Style

#### Background Color

This is used to set the container's background color.

#### Text Color

This is used to set the object's text color.

#### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

#### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

- a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.
- b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### Text Direction

This allows you to set the direction of the text within the object.

### Enable UI

This is enabled for default to allow user interactivity with the object in the UI.

### Is Focusable

Activates / Deactivates focus on the object.

### Unit Converter

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### Enable Touch Manipulation

This is used to activate the object's manipulation options while in runtime.

### Enable Mouse Over

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### Execute Any Enabled Command

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### Show Tooltip When Disabled

Enabling this property will show the tooltip even when the object is disabled.

### Force as Dynamic on Client

Enables/Disables the object's animations on Client side.

### Use IntelliSense

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### Speech Command

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### Preserve Language Font Settings

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### Preserve Style

This is used to preserve the values of the local style property group when the object is updated from the library.

### Preserve code

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### Preserve Colors



This is used to preserve the values of the local color property group when the object is updated from the library.

#### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

#### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

#### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

#### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

#### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

#### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

#### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

#### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

#### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### Access Level from Tag

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### Readable Access Mask

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### Writable Access Area

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Animations

### Animations

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Historian Viewer

The PlatformNext "Historian Viewer" is a configurable object designated to visualizing the historical data of projects recorded with the Historian module.

This Historian Viewer object is available from the "Historical Data Viewers" group in the Toolbox. As difference to the previous "Historical Event Log Viewer" this object loads data from the "UFUAAuditDataItem" table that contains data recorded by the "Historian" being the historical records of variables.

More than one 'Historical Viewer' can be inserted within the project in one or more screens as needed.

# Tag

## Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

## Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

# Historian Viewer Settings

## Layout Configuration

This allows you to configure the object's layout.

## Historian Name

This property is used to select the name of the "Historian " whose data is to be displayed at runtime in the object's grid. This data displayed will only be the recordings of the Tags associating to the selected Historian. When this field is left empty, the grid will display all of the Historian's data, being the recordings of all the Tags associated to different Historians in the project. The data will be filtered based on the time range selected. In addition, during runtime it will always be possible to change the selected Historian to be displayed using the appropriate combo-box on the object's top left.



When the 'Historian Name' property is left empty, the data of all the Historians that use the project's connection string will be displayed. Any other Historian defined its own connection string (hence, record on tables or even different Databases by project default) can only be displayed individually by selecting it in the appropriate field.



The 'Historian Name' property (and the display of Historian data) can be only set when the "Show Audit Trail" property has been disabled.

## DB Connection String

This is used to set the string for the referenced DB connection.

## Filter Type

This is used to a time frame filter for loading data from the database.

## Automatic Update Time

This is used to set the automatic data update time. Leaving this property set with the value = 0, will disable the automatic updating of data. Update time is expresses in seconds.



**Warning!** this feature is only supported in WebHMI Client.

#### Show Best Fit Button

Enabling this will show the best fit button, which is used automatically adjust columns to fit within the Grid object, on the Grid object's top bar and.

#### Show Group Box

Enabling this will show the group box.

#### Show Search Box

Enabling this will show the Search box.

#### Show Filter Box

Enabling this will show the filter box.

#### Show Command Buttons

Enabling this will show the command buttons.

#### Max Rows

This is used to set the maximum number of rows to show in the object.

#### Use Universal Date and Time (UTC)

When enabled, the SourceTimeStamp column will display UTC. When disabled, the local PC time zone will be used.

#### Show Audit Trail

When this property is enabled, the Historian Viewer window will show Audit Trail data of Tags instead of the Project's Historian data.



The data displayed by this window are not subject to validation. In cases when the project's historical data protection has been enabled, the Audit Trail data validation of Tags can only be managed from the "Audit Trail Validator" object.

#### Command Timeout

This is used to set the maximum time to wait for the object to connect to the database in seconds. When left set at 0, timeout will be 30 seconds.

#### User Based Runtime Settings

Enabling this option will allow the object's style to be customized according to the user logged in at runtime.

#### Audit Trail Name

This property, which is shown only when the "Show Audit Trail" property has been enabled, allows the selection of the Tag name subjected to Auditing and whose data will be displayed at runtime in the object's grid. When this field is left empty, the grid will display the data of all the Tags subjected to Auditing. The data will, however be filtered according to the time range selected. In addition, while in runtime you can always change the selected Tag to be displayed by using the appropriate combo-box at the top left of the viewer object's window.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more language defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family

- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

### Row Font Settings

This is used to set the font type for the data column and row contents.

### Header Font Settings

This is used to set which font to use for the Header.

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### **TAB Stop**

This makes the object selectable at runtime using the TAB button.

### **Text Direction**

This allows you to set the direction of the text within the object.

### **Enable UI**

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**



This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project.

This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

#### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

#### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

#### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

#### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

#### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

#### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

#### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**

This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## **Animations**

### **Animations**

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## Commands

### Commands

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## XML System Event Debug

The system log viewer object allows you to view and record information and error messages generated during project edit or runtime mode (loads the data from project log xml file that resides in the project's "Log" folder. It contains information on project starts and stops and any project errors).

For further information also see the topic on: "Finestra system Log"



The data shown in the "System log" window are also saved on file. This file is created in the project's 'LOG' subfolder.

## Tag

### Tag

This allows you to display and assign a variable to the selected object as described in topic: "Assigning Tags to objects".

### Expression and Reverse Expressions

This allows you to set the expression with Excel 2013 syntax whose result will be displayed in the object.

Movicon.NExT supports the possibility to specify and assign an individual bit of a tag or an individual element of an Array tag as explained in topic: "Pointing to Variable Bits and Array Variable Elements" or to define a formula as described in topic: "other expressions"

## Event Debug Settings

### Layout Configuration

This allows you to configure the object's layout.

### Log File Name

This is used to set the path where to save the log file.

### User Based Runtime Settings

Enabling this option will allow the object's style to be customized according to the user logged in at runtime.

## Style

### Background Color

This is used to set the container's background color.

### Text Color

This is used to set the object's text color.

### Horizontal Content Alignment

This is used to horizontally align all the contents within the object.

### Vertical Content Alignment

This is used to vertically align all the contents within the object.

### Disable Anti-Aliasing

This is used to Enable/Disable the Anti-aliasing function.

## Fonts

### Font Size

This is used to set the size of the selected font.

### Font Family

This is used to set a font type to use for text.

### Font Style

This is used to set the style for the select font. The options are:

- Normal
- Oblique
- Italic

### Font Weight

This is used to select the thickness of the selected font.

### Set Font based on Language

This property is used to define a specific Font to be used by the object at runtime for each of the project's languages. In this way, the object will apply the desired Font according to which language is activated.

One or more languages defined in the project can be added to the edit list and each one can be set with a custom Font regarding the following characteristics of:

- Font Family
- Font Size
- Font Style
- Font Weight

When needing to activate a language that has not been defined in the object's FontSettings list, the object's Font will remain with the one already active.

The functionality of this property, in addition to being purely aesthetic, is to allow the most suitable Font for each of the different languages used in the project. For example, text may appear longer in one language than in another and therefore will need a Font that prevents this from happening by using the FontFamily or FontSize most suitable for the different languages.

When deploying the project for WebHMI Runtime, you will need to make sure that the Font associated to the "Font Family" also resides in the target device. To allow the Deploy tool to transfer the Font to the target device, you will need to:

a) copy the Font's .TTF file in the local WebHMI service's "WebHMIServer\wwwroot\fonts" Folder (e.g. "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIServer\wwwroot\fonts"). The "fonts" folder must be created manually when not present.

b) rename the .TTF file so that it reports the "FontName" in full (matching the property exported in the project's "Screen" folder's "<NomeSinottico>.controls" file when deployed).

## Layout

### Position X

This is used to set the object's horizontal position in pixels.

### Position Y

This is used to set the object's vertical position in pixels.

### Height

This is used to change the object's height.

### Width

This is used to change the object's width.

### Border Thickness

This is used to set the container's border thickness.

### Margin

This is used to set the object's margins.

### Lock Object Position

This locks the object's position on screen.

## Visibility

### Transparency

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

### Visibility

This is used to enable the object's visibility.

### Visible on Web Client

This makes the object visible on Web Client.

### Visibility Layer Level

This is used to set the mask that determines the object's visibility.

### Zoom Level Visibility X

This is used to set the zoom factor on the x axis after which the object will become visible on screen.

### Zoom Level Visibility Y

This is used to set the zoom factor on the y axis after which the object will become visible on screen.

## General

This is a list of the general properties which are the same for objects and draws:

### ToolTip

This is used to set the text to be displayed when the user keeps the mouse pointer on the control.



For the objects of the "Trend" category, if the property "**Fit to window**" is set in the synoptic, it will not be possible to display the Tooltip window on WebHMI.

### Menu Activation

This allows you to set the name of the contextual menu that shows when right clicking the object.

### Is Interactive

When enabled, this allows you to interact with the object, or part of it (e.g. the text contained within it).

### Menu with Left button

This enables the contextual menu to show by also using the left mouse key.

### TAB Stop

This makes the object selectable at runtime using the TAB button.

### **Text Direction**

This allows you to set the direction of the text within the object.

### **Enable UI**

This is enabled for default to allow user interactivity with the object in the UI.

### **Is Focusable**

Activates / Deactivates focus on the object.

### **Unit Converter**

This is used to set a converter type for the variables associated to the object. If not already done so, set the Unit Converter resource appropriately before applying this property.

### **Enable Touch Manipulation**

This is used to activate the object's manipulation options while in runtime.

### **Enable Mouse Over**

This is used to enable the "Mouse Over" option to move the mouse on the object during Runtime.

### **Execute Any Enabled Command**

When this property is set with the True value, all the listed enabled commands associated to the button will always be executed. When setting this property with the False value, the commands will only be executed if all are executable. For example, if the list includes a "Tag Value" command type which sets a value in a Tag and the "Execute any Enabled Command" is set to False, the command list will only be executed if the Tag is correctly connected and has Good quality. However, If the property is set at True and the Tag does not have a Good quality, the rest of the commands will be executed all the same.

### **Show Tooltip When Disabled**

Enabling this property will show the tooltip even when the object is disabled.

### **Force as Dynamic on Client**

Enables/Disables the object's animations on Client side.

### **Use IntelliSense**

This is active for default. When disabled, you will need to access variables using "GetVariableValue" and "SetVariableValue". This option is deactivated to improve performances when opening screens.

### **Speech Command**

This is used to assign a speech command to be used at runtime for executing the corresponding action.

### **Preserve Language Font Settings**

This property is used to define a specific Font to be used by the object at runtime for each individual project language. In this way and according to the language that is activated in the project, the object will apply the desired Font set here.

One or more languages defined in the project can be added to the edit list and each one can be customized with its own Font using the following settings:

- Font Family
- Font Size
- Font Style
- Font Weight

In cases where a language is activated but not defined on the object's Language Font List, the object's Font will remain set with the one already activated.

In addition to being purely aesthetic, this function of this property is to allow you to select the Fonts most suited to the different languages used in the project. This is because text in one language may appear longer in another and therefore may need adjusting by using a more appropriate FontFamily or FontSize.

### **Preserve Style**

This is used to preserve the values of the local style property group when the object is updated from the library.

### **Preserve code**

This is used to preserve basic script code of the local style properties when the object is updated from the library.

### **Preserve Colors**

This is used to preserve the values of the local color property group when the object is updated from the library.

### **Preserve Size**

This is used to preserve the values of the local size property group when the object is updated from the library.

### **Preserve Commands**

This is used to preserve the list of commands when the object is updated from the library.

### **Preserve Animations**

This is used to preserve the list of animations when the object is updated from the library.

### **Preserve Menu**

This is used to preserve the set contextual menu, when the object is updated from the library.

### **Preserve Expression**

This is used to preserve the value conversion expression, when the object is updated from the library.

### **Preserve Security**

This is used to preserve the values of the security property group when the object is updated from the library.

### **Preserve Visibility**



This is used to preserve the values of the visibility property group when the object is updated from the library. As regards to project symbols that are created by the user, this option interacts with the "Visibility Level", "Zoom Level Visibility X" and "Zoom Level Visibility Y" properties.

#### **Preserve Tag**

This is used to preserve the connected tag when the object is updated from the library.

#### **Preserve Text**

This is used to preserve the added text string when the object is updated from the library.

## **User Access**

#### **Access Level**

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

#### **User Access Role**

This is used to set the name of the user password group to allow user interaction with this object.

#### **Access Level from Tag**

This is used to set the hierarchical user level requested to allow user interaction with this object by inheriting it from the tag connected to it.

#### **Readable Access Mask**

This is used to set the access mask level in read for the object. The object will not be made visible if the connected user's access mask does not correspond to the one the object has been set with.

#### **Writable Access Area**

This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## **Animations**

#### **Animations**

This property gives access to the window containing dynamic animation commands that can be associated to the object.

## **Commands**

#### **Commands**

This property gives access to the window containing commands that can be associated to this object.

These two property groups are used to access the respective Animations and Commands groups relating to the selected object.



If you cannot see these properties in the Properties Window, make sure that the "Easy Mode" function has been deactivated by using the relevant button on the object's property toolbar.

## Historical Grid

When associated a TAG that has been linked to a Historian resource, the Historical Grid will display data recorded on the relative DV in table or chart format.

To customize the table's graphical aspect, you can use the "EDIT" from the property window to change its size and/or re-order its columns.

For further information, please see: "Historical Trend Server".

## 2.31. Symbol Library Window

### 2.31.1. Using the Symbol Library

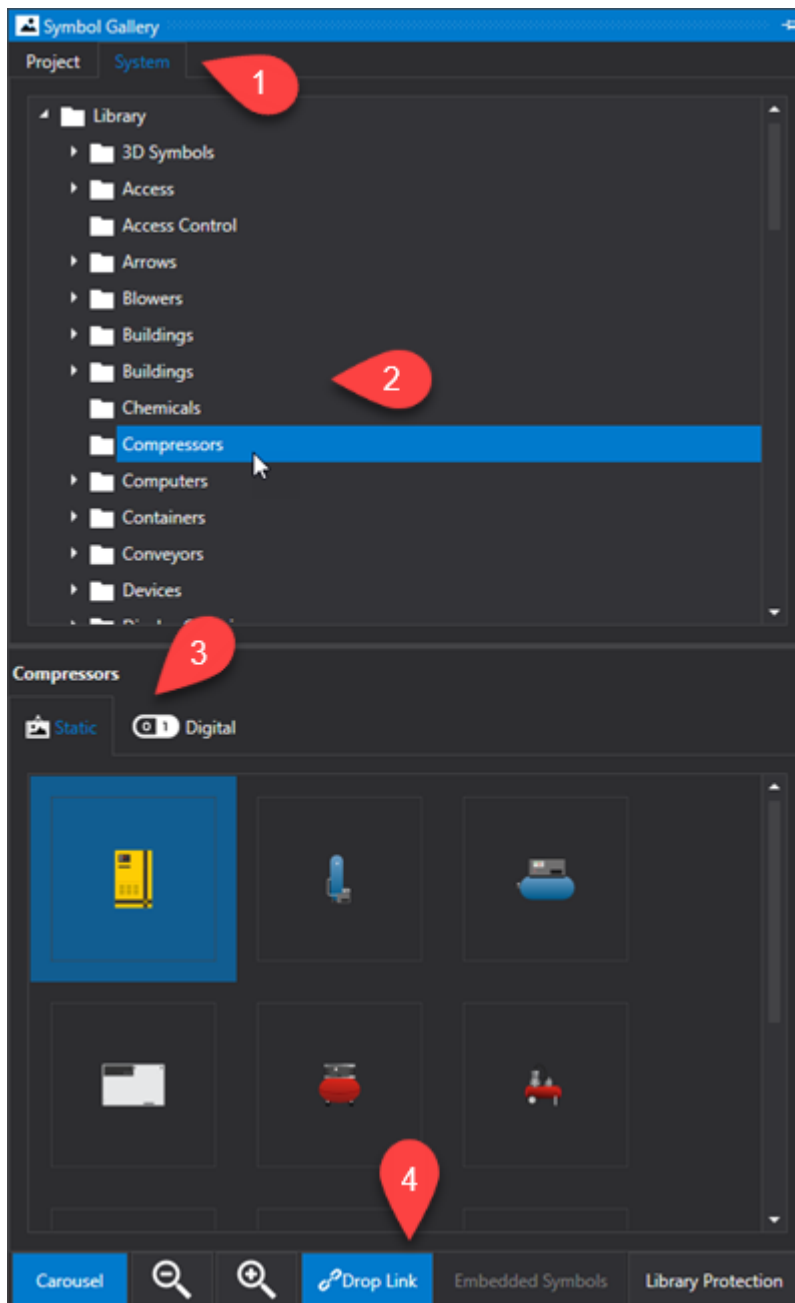
Movicon.NExT offers Symbol Library rich in XAML vector symbols (static and animated) that come predefined and ready to use.

This Symbol Library can be accessed from screens using the relevant TAB on the right side of the workspace.



All the symbols from the Symbol Library can be found in the following link in PDF:

<https://www.progea.com/wp-content/uploads/2020/09/simboli-doc.pdf>



The elements used to build the Symbol Library are:

#### 1. System and Project Tab selection

- System: System Library
- Project: Custom Symbol Library (please see topic on: "Creating Composed Symbols" for further information)



The Movicon.NExT System Symbol Library is the result of industrial design work protected by Copyright. Therefore, the symbols provided by the system library are protected and can only be used for their static or dynamic representation on screens. It is not possible to take the symbols apart or use their XAML source code for any advanced customization. If the user wishes to access the structure of any one of the symbols, they will need to purchase a user license which comes with the unlock code to remove the library's protection.



When the user obtains the library's unlock code, it is important to remember that the symbols remain subject to the User License terms and conditions and Copyright laws. The user must not reveal the protection code or use symbols other than within their own Movicon.NExT projects.

2. Tree structure that lists symbol categories.

3. Static and Animated Tab selection:

- Static: static vector symbols
- Digital: animated vector symbols with digital animation (e.g. value 0 or 1)
- Analog: animated vector symbols with analog animation (e.g. value range from 0 to 100)

4. Library Toolbar:

- Carousel/Grid: previews display of symbols in grid or carousel mode.
- Magnifying glass icon: zooms symbol preview
- Drop Link: allows you to use symbols on screens with links to symbols in the library (parent-child relationship)
- Embedded Symbols: allows you to use symbols on screens as independent symbols (without parent-child relationship)



This option is not available for the system symbol library.

- Library Protection (System Library): This button, if the System Library is active, allows you to decrypt the protected symbols of the system library using the code provided on request.
- Library Protection (Project Library): This button, if the Project Library is activated, allows you to protect / unprotect the personal project library by defining / using a personal password.

### Inserting Symbols into the project

To insert a symbol from the System Library on screen simply:

- Open the Screen of interest in the Movicon.NExT workspace.
- Activate the Symbol Library if not already in the forefront.
- Select the symbol group desired from the window's tree structure list and click on the icon to expand the group.
- Previews of symbols are shown at the bottom of the window. Select the symbol desired and drag it onto the screen.

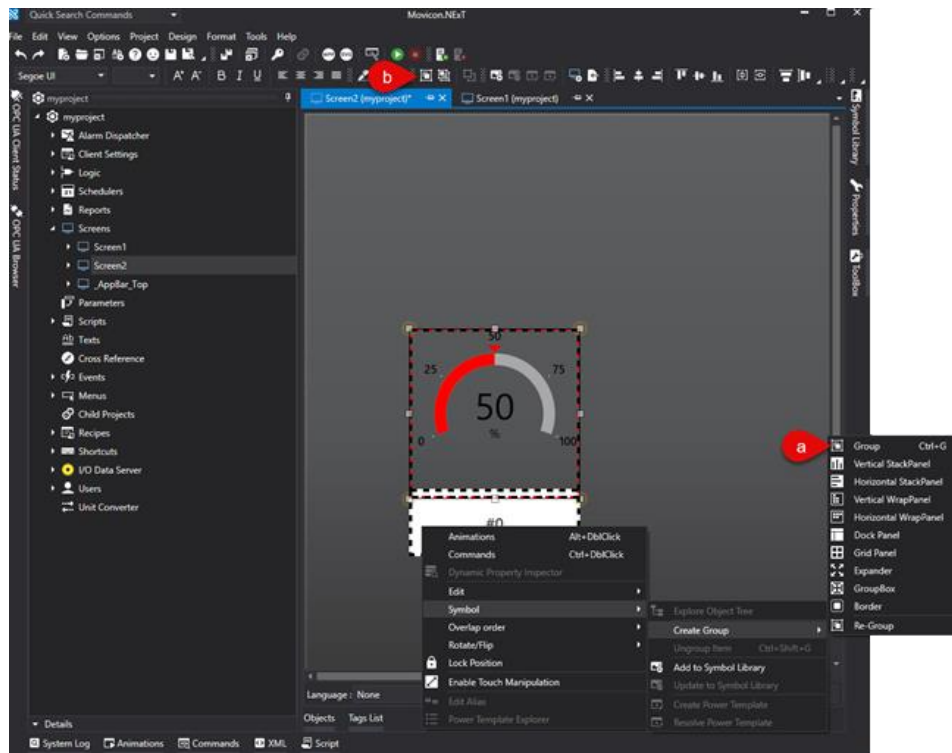
### 2.31.2. Creating Composed Symbols

The graphic object used on screen (User Controls from the toolbox and Symbols from the System's Library) can be grouped together to form a customized symbol composed of a mixed number of elements. This mechanism can result very handy when, for example, the customized symbol must be used many times in the project. These symbols can also be added to the Project's Symbol Library for subsequent reuse.

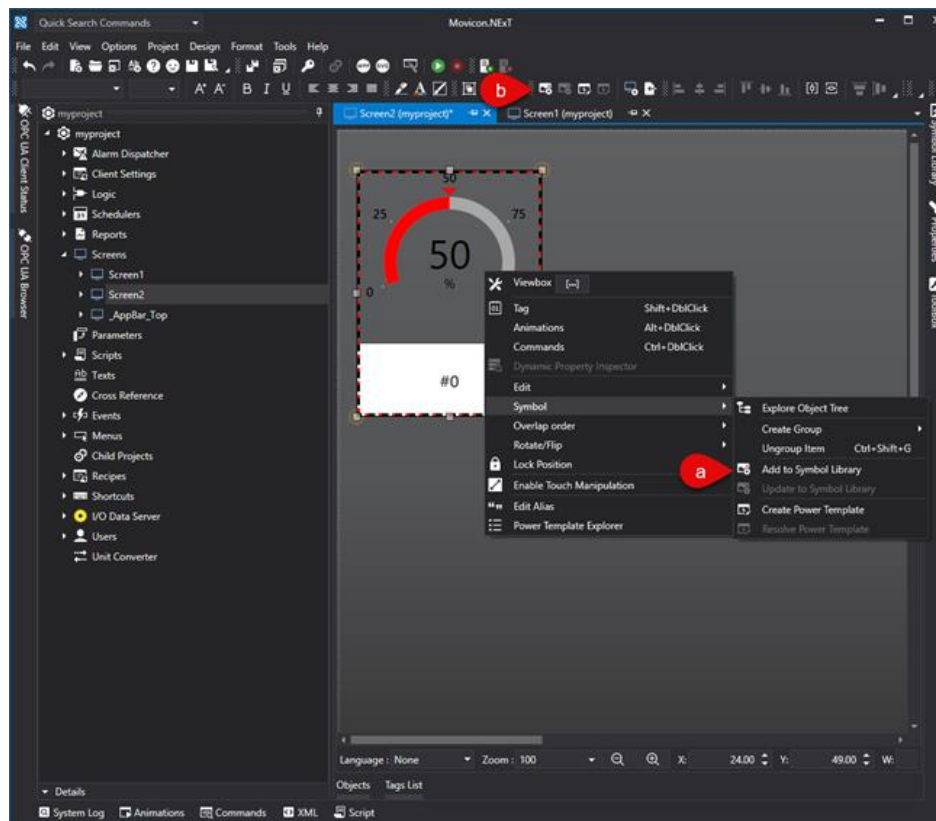
How to create a customized symbol.

To create a customized symbol, simply select a group of graphical objects on screen and then:

- a. Call the contextual menu, which opens when right clicking on the selection of elements, and select Symbol > Create Group > Group.
- b. Alternatively, select the "Group" icon from the Toolbar.
- c. Use the CTRL + G shortcut.

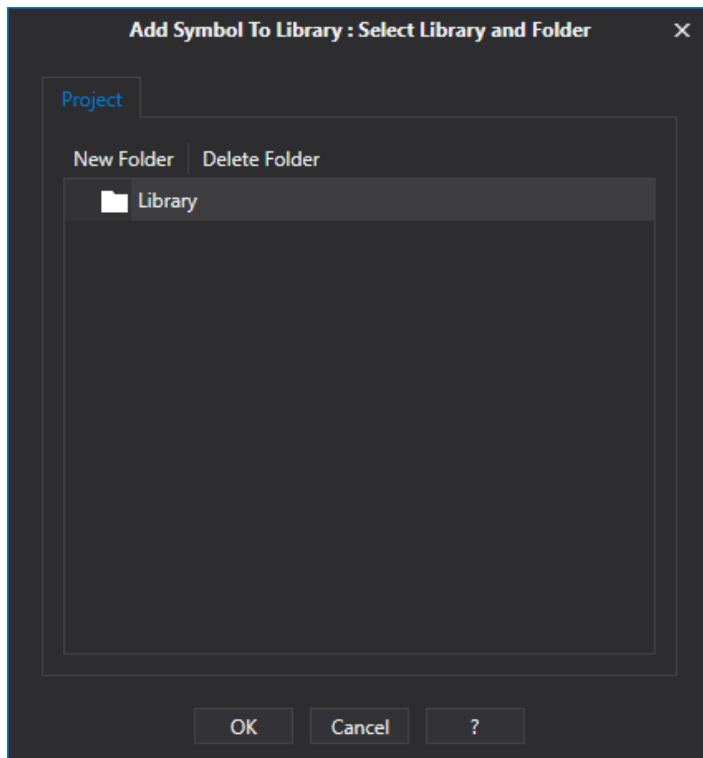


- Once the group has been created, you can insert it in the Project's Symbol Library:
- Call the contextual menu again and select **Symbol > Add to Symbol Library**
  - Or from the Toolbar selecting the "Add To Symbol Library" icon.

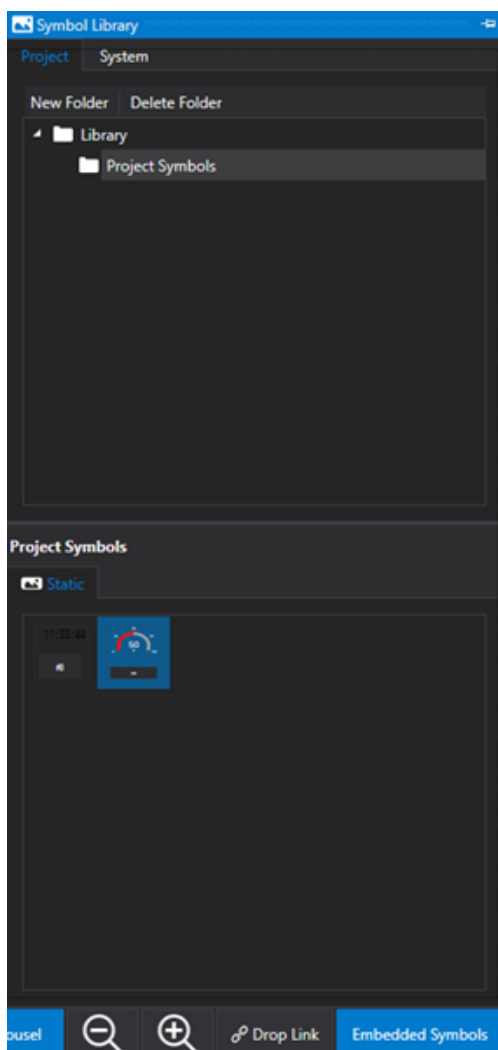


At this point it will be possible to select/create the folder in the Project Library in which to save the custom symbol.

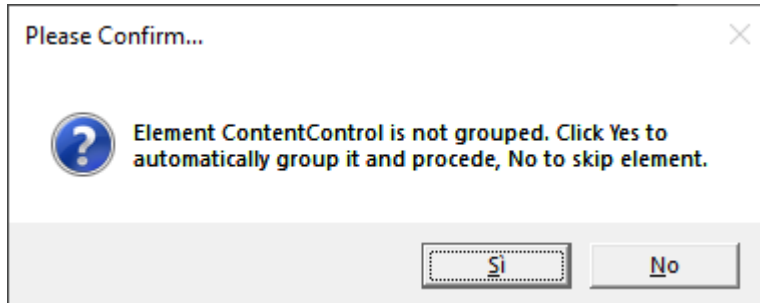
If you do not create/select a folder, a default "Project Symbols" folder will be created once you click the OK button to confirm.



Doing this will also update the symbol library with the new folder.



When creating a custom symbol composed of just one object (e.g. a Display or Button with customized characteristics), you will need to first create a Group to transform the object into a Viewbox. However, in this case, Movicon will prompt a confirm to transform the object into a Viewbox before adding it to the library. In this phase, if more than one object has been selected, each object will be converted and added to the library individually. This operation can be aborted by clicking 'No' in the messagebox:



## Drop Link or Embedded Symbol?

When inserting a symbol from the project's library on screen, you will need to consider the following important concepts that will determine how symbols are managed in the project.

### Drop Link

When dragging a symbol from the library on screen with the "Drop Link" option active, a symbol instance is created on screen which will remain "linked" to the symbol library.

This means that when the corresponding symbol in the library changes, all the symbols' instances in the project will update as a consequence.



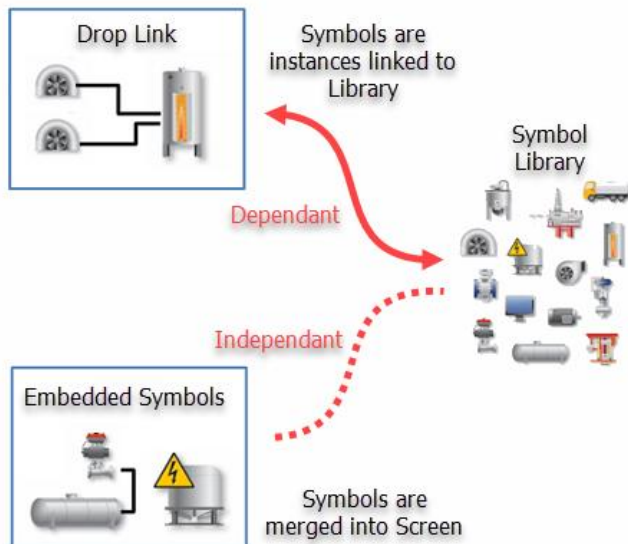
The "Drop Link" option is activated by default

### Embedded Symbol

When dragging a symbol from the library on screen with the "Embedded Symbol" option active, a copy of the symbol is created on screen and therefore it will become local and independent from the Symbol Library.



The "Embedded Symbol" option is not available for the system's library.

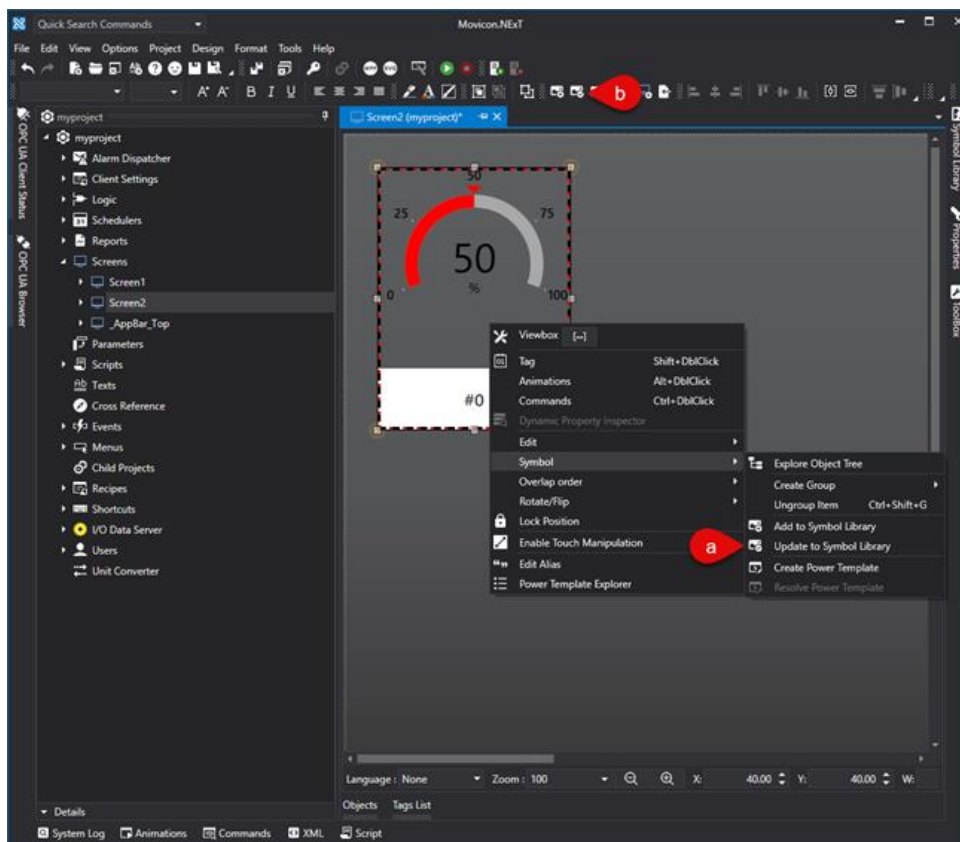


How to edit symbols from the project library.

When using the "Drop Link" option for symbols on screen gives you the advantage to edit the symbol in the library that in turn will be inherited by all they symbols in the project.

To edit/update customized symbols simply follow these steps:

1. Drag the customized symbol on screen with the "Embedded Symbol" option active.
2. Edit the symbol as needed.
3. Call the contextual menu and select Symbol > Update to Symbol Library or (B) select "Update To Symbol Library" from the toolbar.



When you ungroup the symbol, using the Ungroup command, you will need to regroup it by using the "Re-Group" command.

As an alternative, you can edit each single element composing the customized symbol as indicated in the chapter on "Selecting components of Composed Symbols" or by using the Object Explorer window.





The properties that are inherited by symbols in the project following modifications of symbols in the library, are all those properties from the Preserve Template Changes group which are not active.  
For further information, please refer to the "Preserve" properties relating to the objects from the ToolBox.

### Dropping Symbol Code

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

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

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

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

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

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

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

```
    *** Create Props ***
```

```
    If Not Entity.Props.ContainsKey("Prop_Display_MeasUnit") Then
        Entity.Props.Add("Prop_Display_MeasUnit", "mm")
```

```
    End If
```

```
    If Not Entity.Props.ContainsKey("Prop_Display_PrecisionDigit") Then
        Entity.Props.Add("Prop_Display_PrecisionDigit", "1")
```

```
    End If
```

```
    If Not Entity.Props.ContainsKey("Prop_Button_Text") Then
        Entity.Props.Add("Prop_Button_Text", "Increase")
```

```
    End If
```

```
    *****
```

```
    'Force Screen to be Saved
```

```
    Entity.Document.NeedsSave = True
```

```
End Sub
```

### 2.31.3. Power Templates

Power Templates are symbols whose elements have been assigned as variable members of structure variables and to which the create Power Template command has been applied. The Power Templates have specific uses in projects where the same symbol is used as many times as there are structure instances.

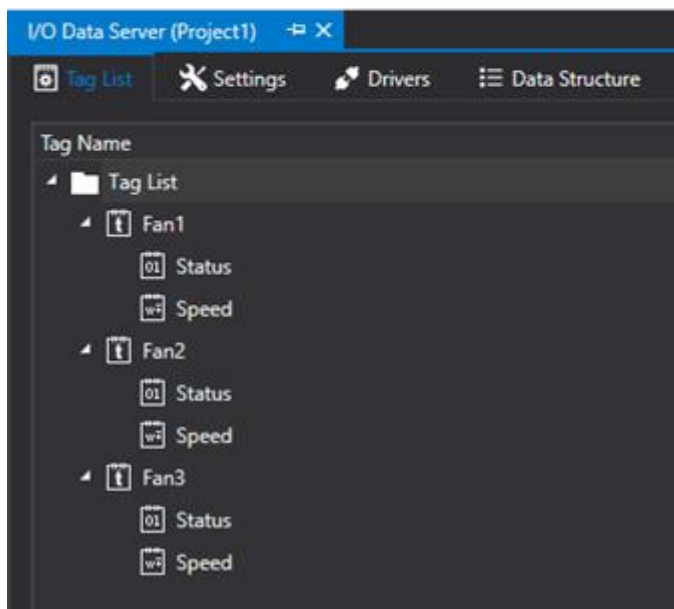
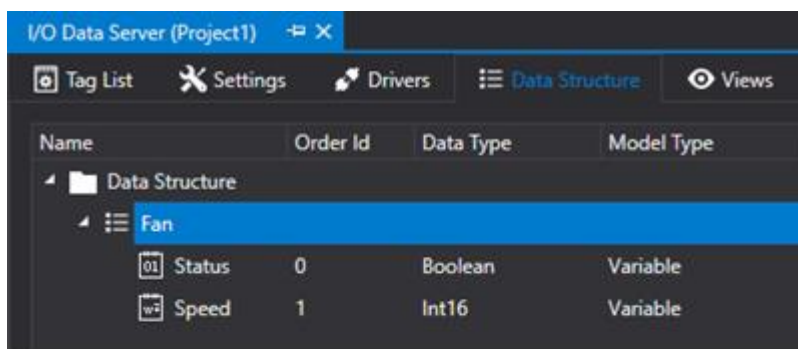


The "Runtime Configuration" management is not supported for viewer objects such as the "Trend" and "Data Analysis" for example, when these objects are used as part of a Power Template.

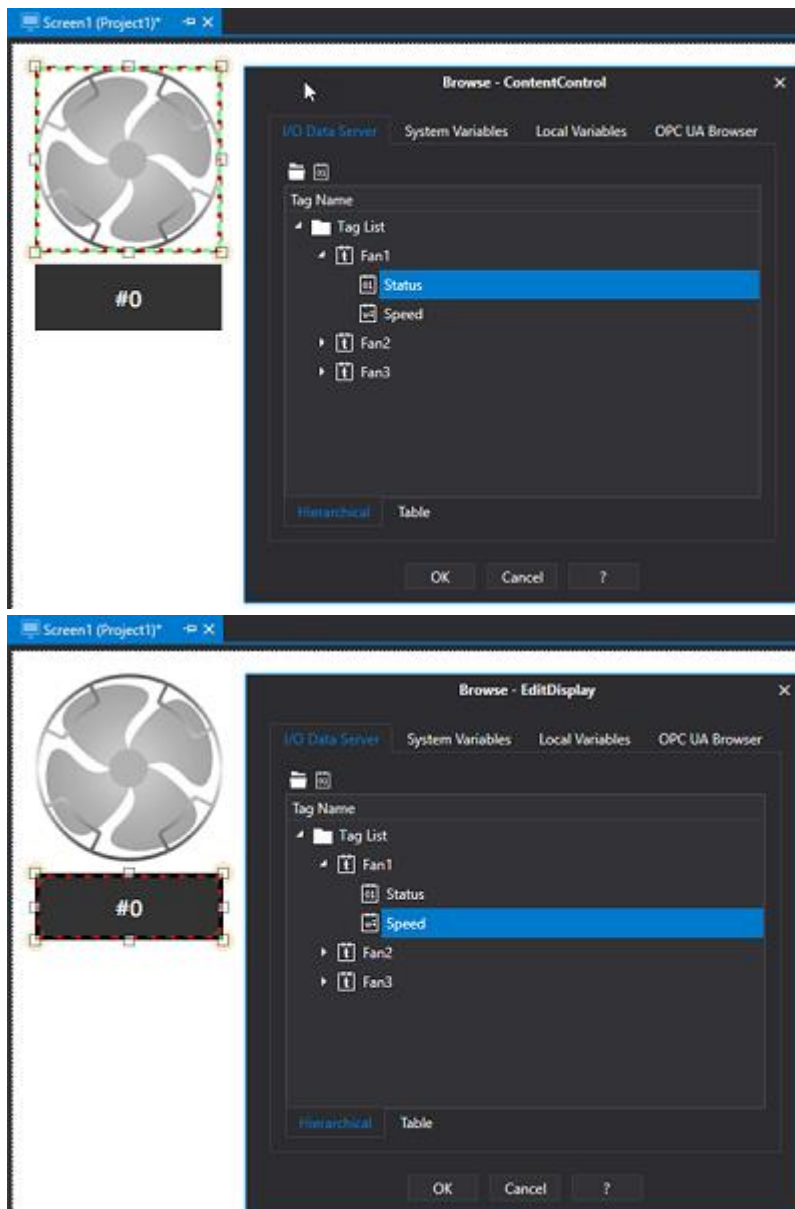
#### Creating a Power Template

To create a power template use the following procedure:

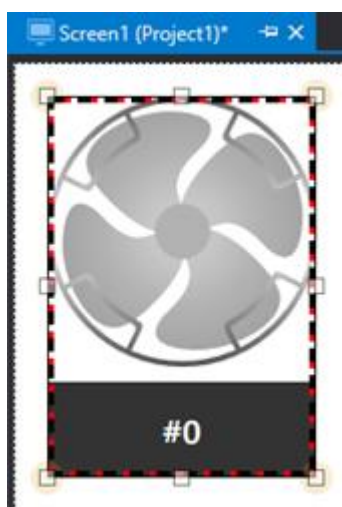
1. Define a Structure with the necessary members.
2. Create one or more Variables associated to the Structure you have just created.



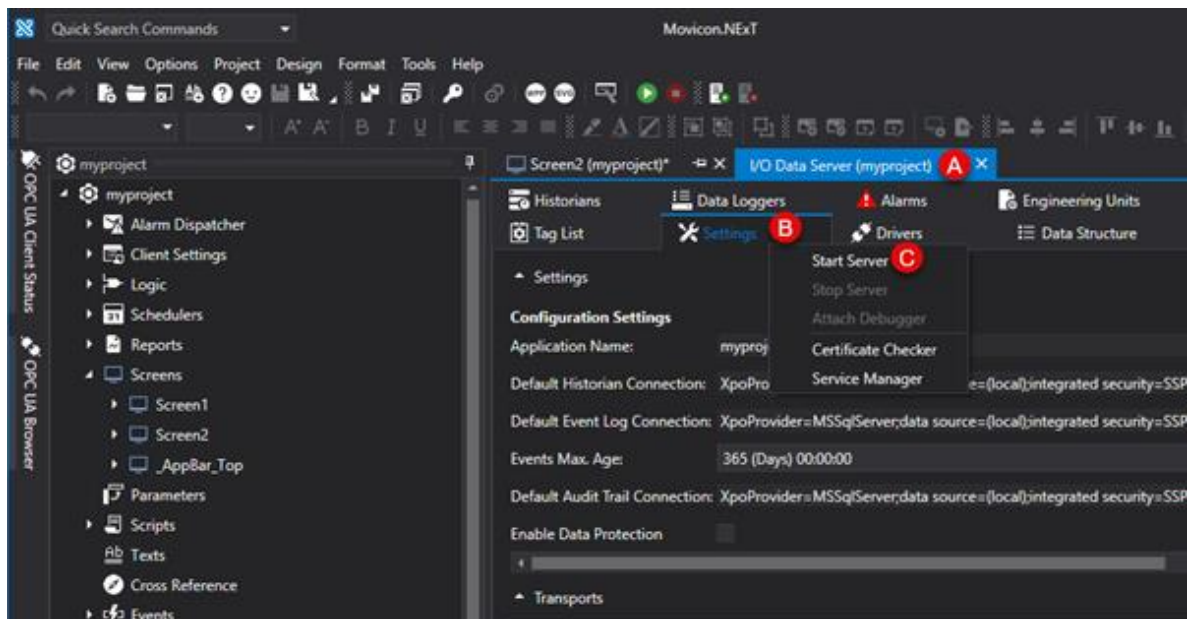
3. Insert the graphical objects you wish to make up the final symbol with on screen and assign them as Tags to one or more members of the Instance Variable (the members should all belong to the same instance Variable).



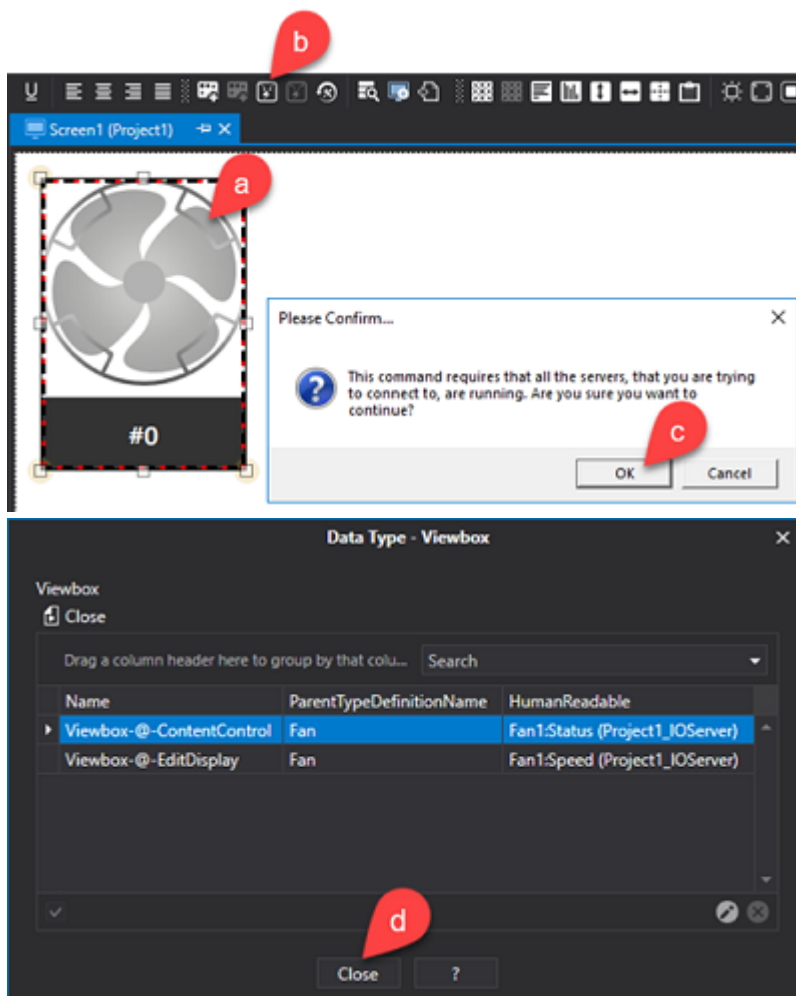
4. Create the whole symbol using the 'Group' command.



5. Startup the project's Server as shown below:

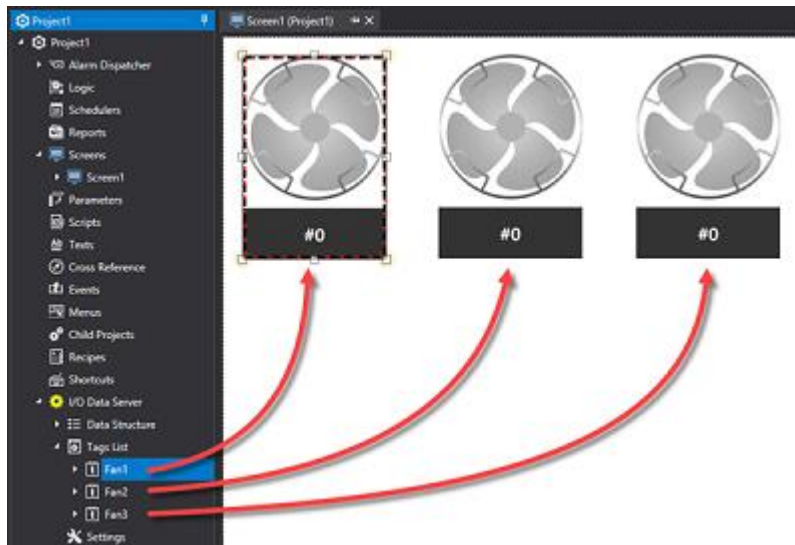


6. Create the Power Template using the "Power Template" command from the Toolbar. Once this operation has terminated, the "Power Template" window will open showing the association between the symbol's objects and the Data Structure. Confirm to close the window.



7. Now add the Power Template you have just created to the Symbol Library using the "Add Symbol to Library" command.

8. The operation to create a Power Template is now completed.
9. Next insert one or more Power Templates on screen and drag a different Instance Variable from those previously created onto each one.



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

In order to for the 'Resolve Power Template' operation to have effect, certain things need to be done. If the Power Template was added to the library as "Merge Code", simply apply the 'Resolve Power Template' command making sure to startup the Server beforehand. If, however, the Power Template was added to the Symbol Library as 'Drop Link', you will first need to make sure that the 'Preserve Tag' property had been checked for each individual Symbol element that has been associated a tag. Once you have done this, startup the Server and apply the 'Resolve Power Template' command.

Once the operation to resolve the Power Template has finished, you can then remove the structure variable associated to the symbol as this will no longer be used.



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

## Power Templates with Server system tags

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



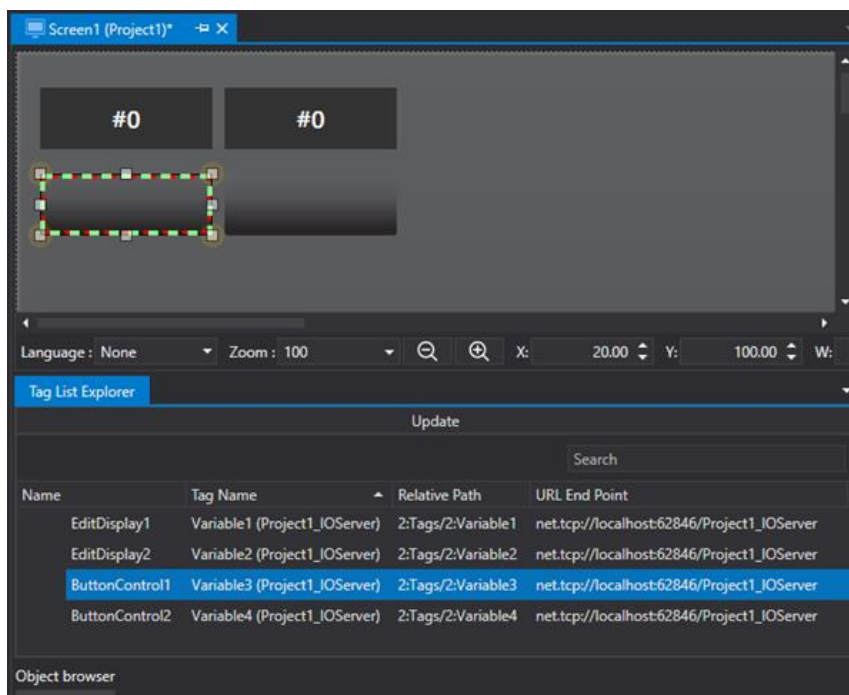
Associating the 'Server' to the template as the default structure, will make the symbol's container object border show in red. This is because the Server does not have a "Quality" property like the Tags do. In order to stop the red border propagating to the symbol's elements as well, you will need to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

## 2.32. Tag List Explorer Window

The "Tag List Explorer Window" can be accessed at the bottom of the screen and allows you to view the tags associated to objects.

When double clicking on a row of the list, the object to which the tag has been associated will automatically select and highlight.

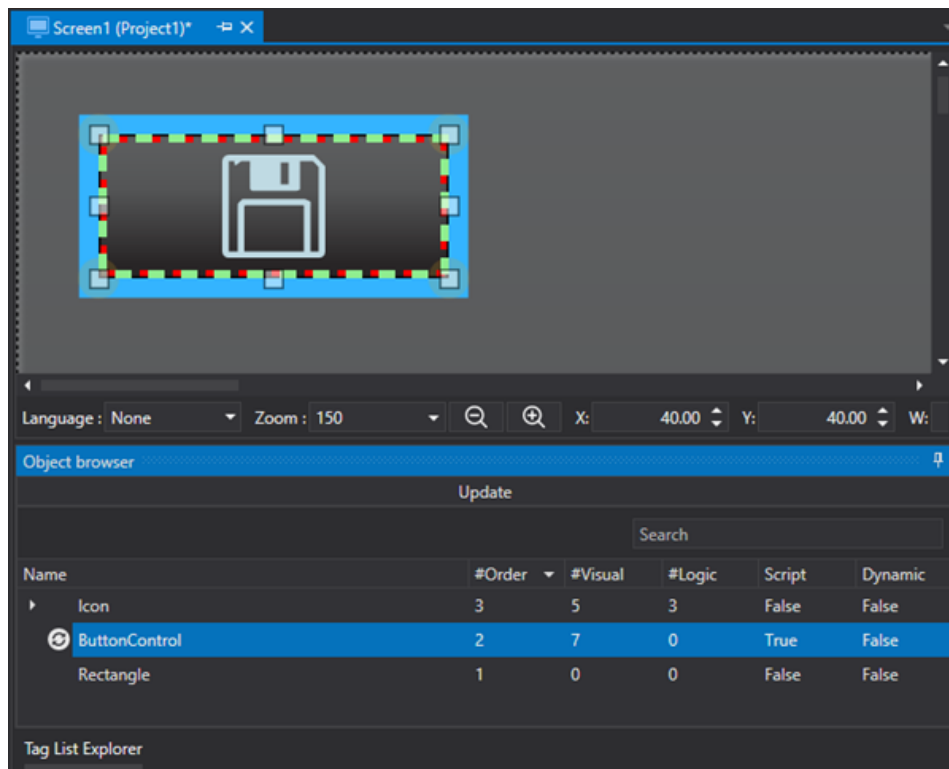
The window's "Update" key will update the Tag list.



## 2.33. Object Explorer Window

The "Object Explorer Window" is accessible at the bottom of the screen and displays graphical objects that are present in the screen.

By double-clicking on a row on the list, the relative object will be automatically selected and highlighted. The "Update" button in the window, updates the list of objects.



The list is ordered according to the order of overlapped objects. Therefore, the row at the top of the list will correspond to the object found at the forefront.

This order can be changed using to the "Order" column.

The window allows users to quickly identify which objects incorporate script code by means of the "Script" column.

## 2.34. Animation Explorer Window

### 2.34.1. Dynamic Animation Settings

Movicon.NExT Object and Symbol animations can be set using the "Animation Explorer" window. This is done by associating Tags that determine the symbol or object's graphical animation during Runtime.



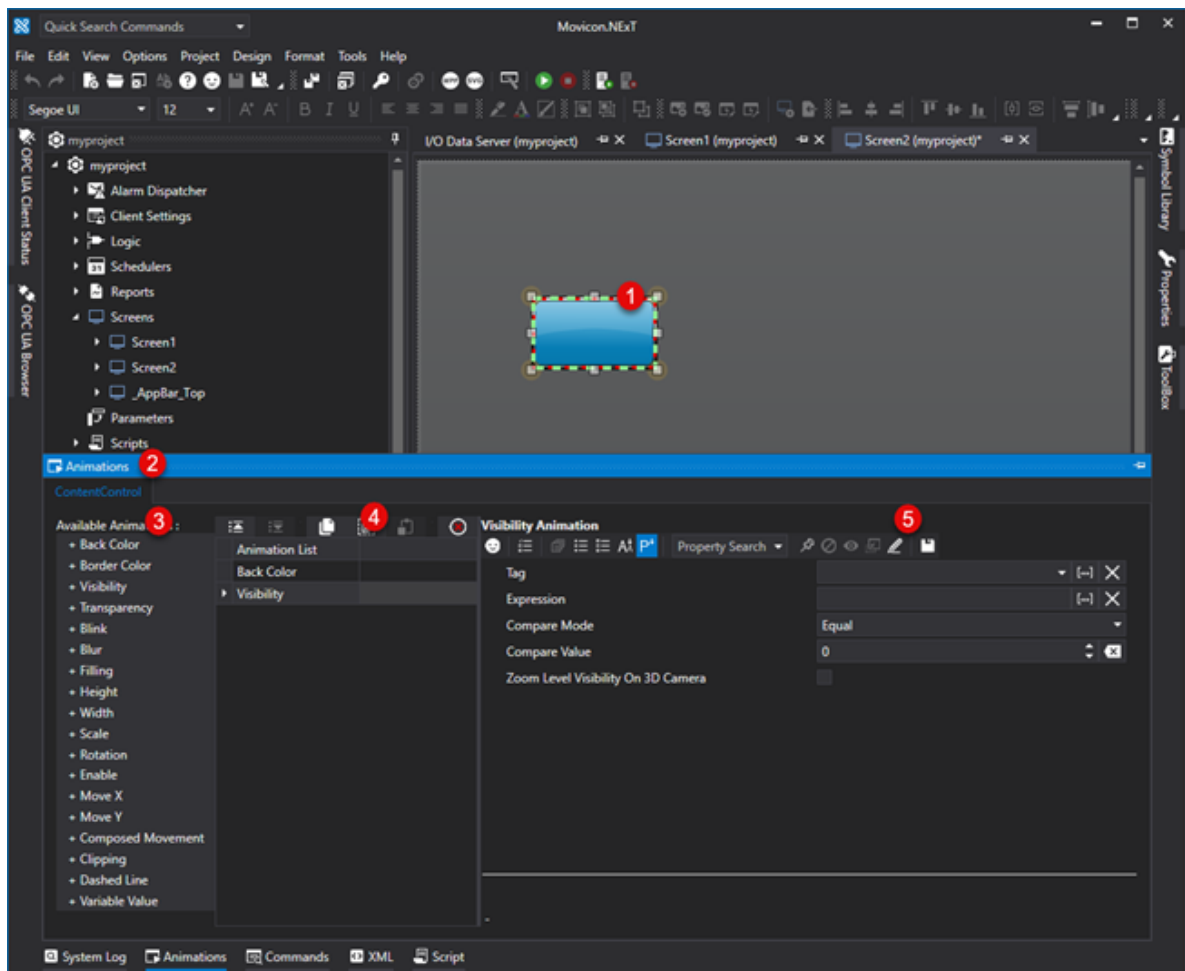
The reference tag name can be specified for each animation inserted in the object by using the object's "Tag" property.

In cases when no Tag has been specified for each individual animation, the object will use the reference Tag it has been associated with (see paragraph: "Assigning Tags to Objects").

### Configuring animations

To configure the animation of an object proceed as follows:





1. Open a screen and select the desired graphical object.
2. Call the "Animation Explorer" window.
3. On the left side of the window there is a list of available animations that can be used. Click the animation you wish to use for the object.
4. Clicking the animation will add it to the list ready for configuring.
5. After having added the desired animation, proceed with configuring its parameters.



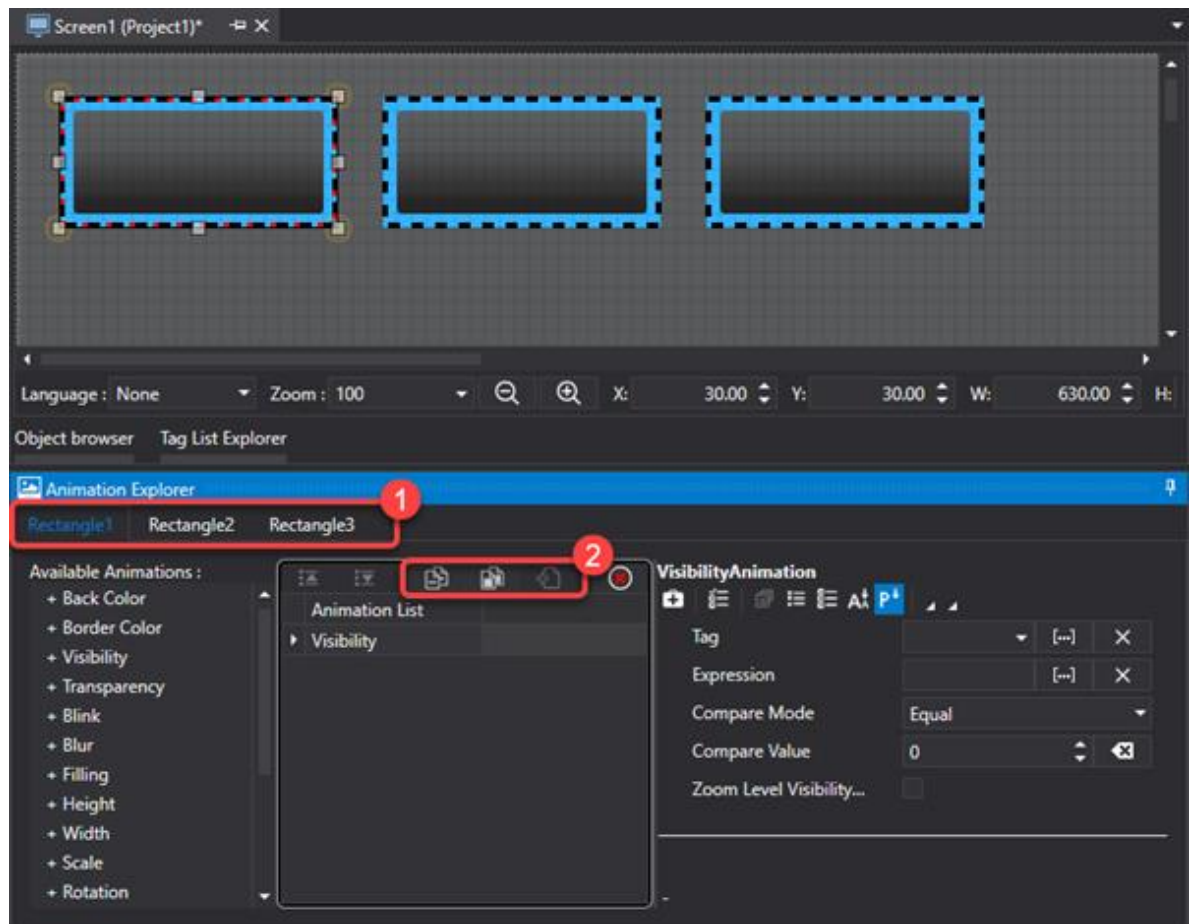
A number of animations can be added to one object.

### Configuring animations in several objects

When multi-selecting a number of objects, you will be able to insert the desired animation in each one.

1. The "Animation Explorer" window will open with as many Tabs as there are selected objects (each one with the name of the object). When selecting one of the Tabs, the Animation List of the object represented by the Tag will activate for editing.
2. Animations can be copied from one object to the next by using the Copy, Copy All and Paste buttons.





### 2.34.2. Back Color Animation

The Dynamic "Back Color" Animation command is used for assigning a variable to the object to determine the coloring of the object's background. This command requires that the relative configuration parameters be defined along with colors relating to the value thresholds.

The function's behaviour is conditioned by the setting of the properties described in "Dynamic Animation Properties".

## Back Color Animation Properties

### Animation Tag

This sets the tag to associate to the animation.

### Expression

This is used to set the expression whose result will be used in the animation.



Certain syntax can be used within this property for the following scopes:

- to connect the nth Bit of a Tag, insert the '.nBit' value.

(e.g. .1 to indicate Bit 1)

- to connect the nth element of an array, insert the '[nElem]' value.

(e.g. [1] to indicate the item 1)

### Apply this animation to all the group's objects

If the object is part of an object group created by the user, this property will extend the animation to all the group's objects.

### Color Configuration

**Value:** defines which Tag value the specified color refers to.

**Color:** defines the color to be applied when the Tag assumes the specified value.

**Blink Time:** Defines the blink duration (in milliseconds) between the main Color and the Blink Color.

**Blink Color:** Defines the color to alternate with the main color.

### 2.34.3. Border Color Animation

The Dynamic "**Border Color**" Animation command is used for assigning a variable to the object to determine the color of the object's borders. This command requires that the relative configuration parameters be defined along with the colors relating to the value thresholds.

The function's behaviour is conditioned by the property settings described in "Dynamic Animation Properties".

## Border Color Animation Properties

### Animation Tag

This sets the tag to associate to the animation.

### Expression

This is used to set the expression whose result will be used in the animation.



Certain syntax can be used within this property for the following scopes:

- to connect the nth Bit of a Tag, insert the '.nBit' value.

(e.g. .1 to indicate Bit 1)

- to connect the nth element of an array, insert the '[nElem]' value.

(e.g. [1] to indicate the item 1)

### Apply this animation to all the group's objects

When the object is part of an object group created by the user, this property consents the animation to be applied to all the objects in that group.

## Color Configuration

**Value:** Defines which Tag value the color refers to.

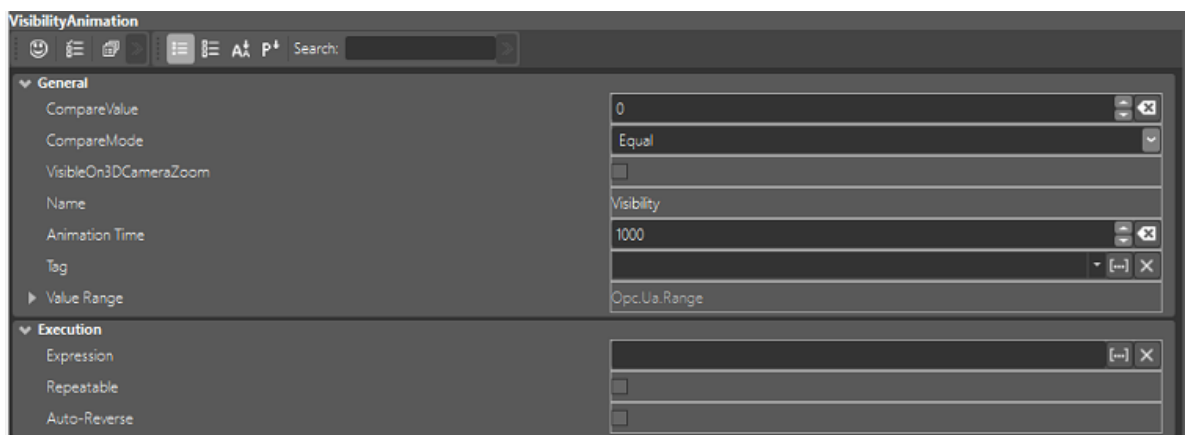
**Color:** Defines the color to apply when the Tag assumes the specified Tag.

**Blink Time:** Defines the blink duration (in milliseconds) between the main color and the blink color.

**Blink Color:** Defines the color to alternate with the main Color.

## 2.34.4. Visibility Animation

The Dynamic **Visibility** Animation command allows you to assign objects with a variable whose value will determine their visibility.



## Visibility Animation Properties

### Animation Tag

This sets the tag to associate to the animation.

### Expression

This is used to set the expression whose result will be used in the animation.



Certain syntax can be used within this property for the following scopes:

- to connect the nth Bit of a Tag, insert the '.nBit' value.

(e.g. .1 to indicate Bit 1)

- to connect the nth element of an array, insert the '[nElem]' value.

(e.g. [1] to indicate the item 1)

### Compare Value

The reference value to determine the object's visibility or invisibility is entered in this field. This value will be compared with the value of the tag assigned to the animation as a reference tag or inherited from the symbol according to the selection made in the 'Compare Mode' field.

## Compare Mode

this field is used to select the compare mode with which the object's visibility will depend on. The comparison modes are:

- Equal
- Major
- Minor
- MajorEqual
- MinorEqual

A comparison will be performed on the 'Compare Value' field and the value contained in the reference Tag.

## Zoom Level Visibility On 3D Camera

this property has effect only when the visibility animation is applied to a 3D symbol element and not to the symbol as a whole. When this option is enabled, the element's visibility will be management exclusively according to the zoom level executed in the object's 3D camera (right click + drag object). In addition, there is no need to define a Tag to use as a reference and the Compare Mode will be ignored. In this case, the element will become visible when the object's scale level is lower than the value indicated in the 'Compare Value'. Take into account that the value 1 represents the 3D object's initial scale value. For example, if you set the value = 2, the element will become visible only when the 3D symbol reaches a zoom value that is twice the initial size.

## 2.34.5. Transparency Animation

The Dynamic "**Transparency**" Animation command is used to assign a variable to the object to determine the object's transparency and visibility values according to the following parameters:

## Animation Transparency Visibility Properties

### Animation Tag

This sets the tag to associate to the animation.

### Expression

This is used to set the expression whose result will be used in the animation.



Certain syntax can be used within this property for the following scopes:

- to connect the nth Bit of a Tag, insert the '.nBit' value.

(e.g. .1 to indicate Bit 1)

- to connect the nth element of an array, insert the '[nElem]' value.

(e.g. [1] to indicate the item 1)

### Animation Behaviour

Defines the animation behaviour based on these options:

- **Absolute:** The variable's max. value range represents the final result of the set animation expressed with the measure unit compatible with the animation to which it refers.
- **Proportional:** The variable's max. value range is proportional to the animation's final value.
- **Trigger:** The variable value is handled as a "trigger", therefore with value  $\geq 1$ , animation will occur at the final value established.

### Value Range

Sets the value range allowed for managing the animation.

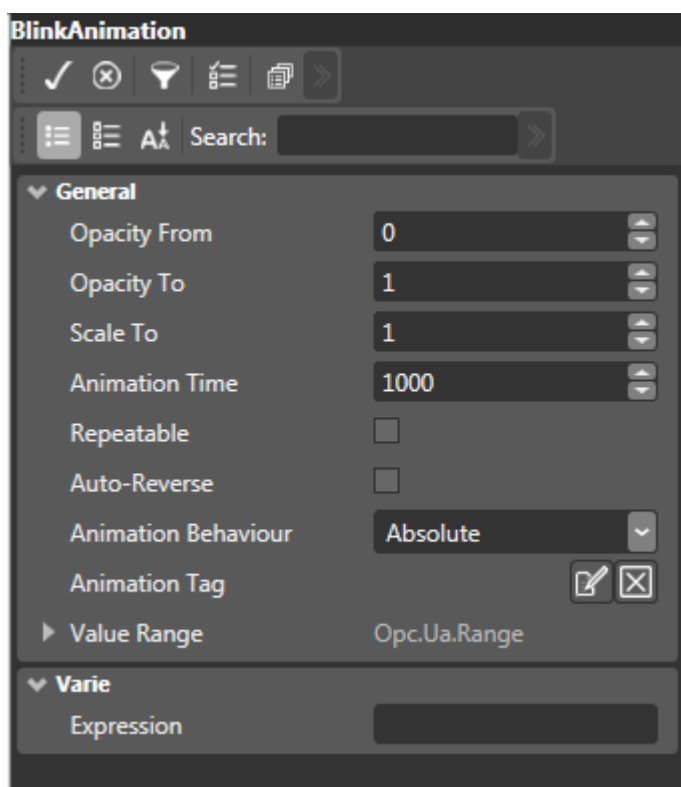
- **Tag Min Value**
- **Tag Max Value**

### Transparency

This is used to define the object's visibility rate by defining its transparency level in respect to its original visibility rate. The value = 0 determines maximum transparency (INVISIBLE), while the value = 1 determines maximum visibility.

## 2.34.6. Blink Animation

The Dynamic "**Blink**" Animation command is used for assigning a variable to the object to determine visual blink animation by managing a scaled zoom and a transitory transparency value according to the following parameters:



## Blink Animation Properties

### Animation Tag

This sets the tag to associate to the animation.

### Expression

This is used to set the expression whose result will be used in the animation.



Certain syntax can be used within this property for the following scopes:

- to connect the nth Bit of a Tag, insert the '.nBit' value.

(e.g. .1 to indicate Bit 1)

- to connect the nth element of an array, insert the '[nElem]' value.

(e.g. [1] to indicate the item 1)

### Animation Time

This is used to define the coefficient time (in milliseconds) in the animation management to make the dynamic animation smoother.

### Min Transparency

Defines the object's minimum transparency value, where the value = 0 means "invisible"

### Max Transparency

Defines the object's maximum transparency value, where the value = 1 means "Full Visibility".

### Object Scale

This is used to define a scale value while blinking, where the value = 1 (default) refers to the object's natural scale size (100%).

## 2.34.7. Blur Animation

The Dynamic "**Blur**" Animation allows a tag to be assigned to the object that will determine the object's blur effects according to tag variation.

## Blur Animation Properties

### Animation Tag

This sets the tag to associate to the animation.

### Expression

This is used to set the expression whose result will be used in the animation.



Certain syntax can be used within this property for the following scopes:

- to connect the nth Bit of a Tag, insert the '.nBit' value.

(e.g. .1 to indicate Bit 1)

- to connect the nth element of an array, insert the '[nElem]' value.

(e.g. [1] to indicate the item 1)

### Animation Behaviour

This is used to set the animation management mode in respect to the value range set in the "Value Range" property, for the TAG connected to the animation. The values can be:

- **Absolute:** the object continues to blink between minimum transparency value set and the reference tag value. In this case, if the set transparency is Min=0 and Max=1, when the reference tag has a 0 value, the object will blink with a transparency between 0 and 0 and will result invisible. When the tag is equal to one, the object will blink with a transparency between 0 and 1. When the Tag is 0.5, the object will blink with a transparency between 0 and 0.5.
- **Proportional:** the object blinks between the minimum transparency value and the referenced tag's "scaled" value. Scaling is done between the set tag range, e.g. 0-100 and the transparency range eg. 0-1. In this case, if the set transparency is Min=0 and Max=1, when the reference tag has a zero value, the object will blink with a transparency between 0 and 0 and will result invisible. When the tag is equal to 100 the object will blink with at transparency between 0 and 1. When the Tag is 50, the object will blink with a transparency between 0 and 0.5.

### Blur Radius

This property is only available for proportional animation types. It is used to set the blur radius to be applied.

### Value Range

This property is only available for proportional animation types. It is used to the value range consented for managing animation.

- **Min Tag Value**
- **Max Tag Value**

## 2.34.8. Filling Animation

The Dynamic Filling Animation command is used for assigning a variable to the object to determine the dynamic filling of objects such as polygons according to the following parameters:

### Animation Tag

This sets the tag to associate to the animation.

### Expression

This is used to set the expression whose result will be used in the animation.



Certain syntax can be used within this property for the following scopes:

- to connect the nth Bit of a Tag, insert the '.nBit' value.

(e.g. .1 to indicate Bit 1)

- to connect the nth element of an array, inset the '[nElem]' value.

(e.g. [1] to indicted the item 1)

### Animation Time

Defines the coefficient animation time in milliseconds to make the dynamic animation flow smoothly.

### Animation Behaviour

Defines the animation's behaviour according to these two options:

- **Absolute:** the variable's maximum range value represents the set animation's end result using the measure unit that is compactible with the animation it refers to.
- **Proportional:** the variable's maximum range value is proportional to the animations end value.

### Value Range

This property is only available with proportional type animations. It sets value range allowed for managing the animation:

- **Tag Min Value**
- **Tag Max Value**

### Repeatable

This is used for enabling/disabling animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active.

### Start Color

This is used for selecting a starting color to be used in the filling process.

### End Color

This is used for selecting a color to end the filling process.

### Filling Type

This is used for selecting the direction of the filling animation.

### Filling Offset

This is used for defining an "offset" to the gradient color factor value (1 = Default value)

### Gradient Type

This is used for selecting which type of gradient to use between linear or radial for the colour filling.

### Radial Radius X

This is used for defining the Radius X value when selecting the radial gradient type.

### Radial Radius Y

This is used for defining the Radius X value when selecting the radial gradient type.

## 2.34.9. Height Animation

The "**Height**" Animation command permits a variable to be assigned to the object to determine the object's height size (in pixels) according to the following parameters:

### Animation Tag

This sets the tag to associate to the animation.

### Expression

This is used to set the expression whose result will be used in the animation.



Certain syntax can be used within this property for the following scopes:



- to connect the nth Bit of a Tag, insert the '.nBit' value.

(e.g. .1 to indicate Bit 1)

- to connect the nth element of an array, insert the '[nElem]' value.

(e.g. [1] to indicate the item 1)

### Animation Time

Defines the time coefficient in milliseconds to manage animation making it dynamically smoother.

### Animation Behaviour

Defines the animation behaviour based on these options:

- **Absolute:** The variable's max. value range represents the final result of the set animation expressed with the measure unit compatible with the animation to which it refers.
- **Proportional:** The variable's max. value range is proportional to the animation's final value.

### Value Range

Sets the value range allowed for managing the animation.

- **Tag Min Value**
- **Tag Max Value**

### Repeat

Enables animation repetition. When active, the executed animation will be repeated in a loop for all the time active.

### Height

Defines the height size value in pixels to reach. The behaviour of this function is conditioned by the property settings described above.

## 2.34.10. Width Animation

The Dynamic "**Width**" Animation is used for assigning a variable to the object that will determine the object's width size in pixels according to the following parameters:

## Width Animation Properties

### Animation Tag

This sets the tag to associate to the animation.

### Expression

This is used to set the expression whose result will be used in the animation.



Certain syntax can be used within this property for the following scopes:

- to connect the nth Bit of a Tag, insert the '.nBit' value.

(e.g. .1 to indicate Bit 1)

- to connect the nth element of an array, inset the '[nElem]' value.

(e.g. [1] to indicted the item 1)

### Blink Time

This is used to define the coefficient time (in milliseconds) in the animation management to make the dynamic animation flow more smoothly.

### Animation Behaviour

Defines the animation behaviour based on these options:

- **Absolute:** The variable's max. value range represents the final result of the set animation expressed with the measure unit compatible with the animation to which it refers.
- **Proportional:** The variable's max. value range is proportional to the animation's final value.

### Value Range

Sets the value range allowed for managing the animation.

- **Tag Min Value**
- **Tag Max Value**

### Repeatable

This is used to enable/disable animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active.

### Width value

This is used to define the width size in pixels to be reached. The behaviour of this function is conditioned by the settings in the properties described in the paragraph entitled.

## 2.34.11. Scale Animation

The Dynamic "**Scale**" Animation command is used for assigning a variable to the object to determine the scale size percentage according to the following parameters:

## Scale Animation Properties

### Animation Tag

This sets the tag to associate to the animation.

### Expression

This is used to set the expression whose result will be used in the animation.



Certain syntax can be used within this property for the following scopes:

- to connect the nth Bit of a Tag, insert the '.nBit' value.

(e.g. .1 to indicate Bit 1)

- to connect the nth element of an array, insert the '[nElem]' value.

(e.g. [1] to indicate the item 1)

## Animation Time

**Defines the time coefficient (in milliseconds) in managing animation by rendering it dynamically smoother.**

### Animation Behaviour

Defines the animation behaviour based on these options:

- **Absolute:** The variable's max. value range represents the final result of the set animation expressed with the measure unit compatible with the animation to which it refers.
- **Proportional:** The variable's max. value range is proportional to the animation's final value.

### Value Range

Sets the value range allowed for managing the animation.

- **Tag Min Value**
- **Tag Max Value**

### Repeatable

This is used for enabling/disabling animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active.

### Auto-Reverse

This is used for enabling/disabling the auto-reverse. When enabled the animation will reverse in a loop until the "Animation Condition" remains active.

### Scale Factor Value

This property is available only with the proportional animation types. It is used to define the percentage rate of the object's scale size in respect to its original size. The value = 1 represents the object's original size.

### Scaling Type

Sets which axis the scale is to be applied on.

### 2.34.12. Rotation Animation

The Dynamic "**Rotation**" Animation command allows you to assign a variable to the object that will determine the angle rotation of the object in respect to its barycenter, according to the following parameters:



The default rotation barycenter is in the object's center, but can be changed in graphics editing mode by activating the display of the barycenter's drag point by using the relevant item from the object's command menu.

## Angle Rotation Animation Properties

### Animation Tag

This sets the tag to associate to the animation.

### Expression

This is used to set the expression whose result will be used in the animation.



Certain syntax can be used within this property for the following scopes:

- to connect the nth Bit of a Tag, insert the '.nBit' value.

(e.g. .1 to indicate Bit 1)

- to connect the nth element of an array, insert the '[nElem]' value.

(e.g. [1] to indicate the item 1)

### Animation Time

This is used to set the co-efficiency time, in milliseconds, in the animation's management to make the animation smoother.

### Animation Behaviour

Defines the animation behaviour based on these options:

- **Absolute:** The variable's max. value range represents the final result of the set animation expressed with the measure unit compatible with the animation to which it refers.
- **Proportional:** The variable's max. value range is proportional to the animation's final value.

### Value Range

Sets the value range allowed for managing the animation.

- **Tag Min Value**
- **Tag Max Value**

### Repeatable

This is used for enabling/disabling animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active.

### Auto-Reverse

This is used for enabling/disabling the auto-reverse. When enabled the animation will be executed in the reverse until this condition remains active.

### Rotation Angle

This property is only available for proportional animation types. It is used to define the angle position to reach. The functions behaviour is conditioned by the property settings described here.

## 2.34.13. Enable Animation

The Dynamic "**Enable**" Animation command is used for assigning a variable to the object so that it can be enabled for activation or manipulation. This type of animation is referred to the objects that invoke commands (e.g. buttons).

The associated variable determines whether to enable or not to enable the object's activation to then permit the execution of any associated commands.

## Enable Animation Properties

### Animation Tag

This sets the tag to associate to the animation.

### Expression

This is used to set the expression whose result will be used in the animation.



Certain syntax can be used within this property for the following scopes:

- to connect the nth Bit of a Tag, insert the '.nBit' value.

(e.g. .1 to indicate Bit 1)

- to connect the nth element of an array, insert the '[nElem]' value.

(e.g. [1] to indicate the item 1)

### Comparison Mode

The reference value to determine whether the object should be made visible or not is entered in this field. This value will be compared, according to the selection made in the 'Compare Mode' field, with the value of the reference tag assigned to the animation or inherited by the symbol.

### Comparison Value

In this field you should select the comparison mode whose result the object's visibility will depend on. The possible comparison types are:

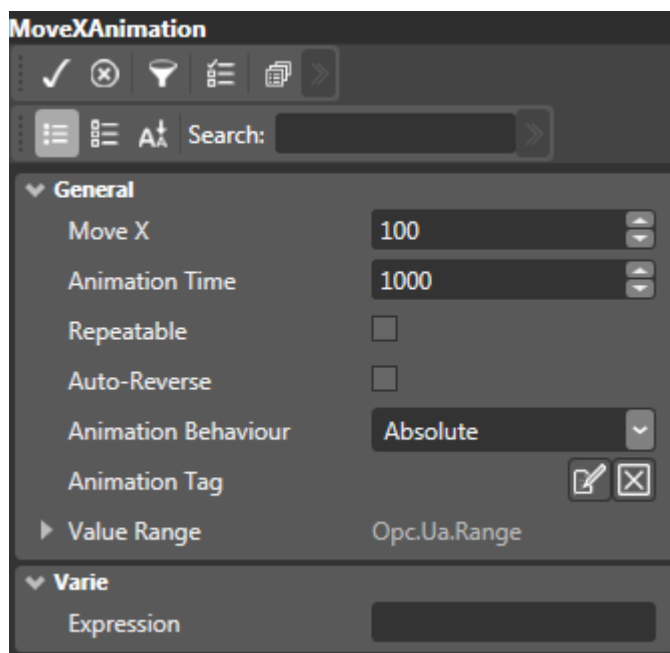
- Equal

- Major
- Minor
- Major and Equal
- Minor or Equal

The comparison will be made between the value defined in the 'Compare Value' field and the value contained in the reference Tag.

#### 2.34.14. Horizontal X Movement Animation

The "**Move X**" Dynamic Animation command is used for assigning a variable to the object to determine the object's position using a horizontal movement along the X axis according to the following parameters:



### Move X Animation Properties

#### Animation Tag

This sets the tag to associate to the animation.

#### Expression

This is used to set the expression whose result will be used in the animation.



Certain syntax can be used within this property for the following scopes:

- to connect the nth Bit of a Tag, insert the '.nBit' value.

(e.g. .1 to indicate Bit 1)

- to connect the nth element of an array, insert the '[nElem]' value.

(e.g. [1] to indicate the item 1)

### Animation Time

Defines the coefficient animation time in milliseconds to make the dynamic animation flow smoothly.

### Animation Behaviour

Defines the animation's behaviour according to these two options:

- **Absolute:** the variable's maximum range value represents the set animation's end result using the measure unit that is compactible with the animation it refers to.
- **Proportional:** the variable's maximum range value is proportional to the animations end value.

### Value Range

This property is only available with proportional type animations. It sets value range allowed for managing the animation:

- **Tag Min Value**
- **Tag Max Value**

### Repeatable

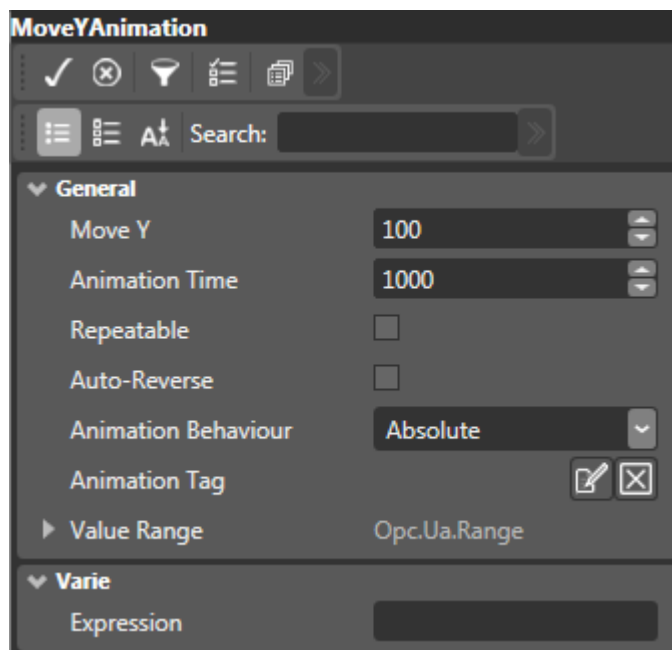
This enables animation repetition. When active, the animation will repeat cyclically for the entire time this options remains active.

### Horizontal Movement

Defines the object's movement value on the X axis in pixels, in respect to its starting point.

## 2.34.15. Vertical Y Movement Animation

The "**Move Y**" Dynamic Animation command is used to assign a variable to the object to determine its position by using a vertical movement along the Y axis according to the following parameters:



## Move Y Animation Properties

### Animation Tag

This sets the tag to associate to the animation.

### Expression

This is used to set the expression whose result will be used in the animation.



Certain syntax can be used within this property for the following scopes:

- to connect the nth Bit of a Tag, insert the '.nBit' value.

(e.g. .1 to indicate Bit 1)

- to connect the nth element of an array, insert the '[nElem]' value.

(e.g. [1] to indicate the item 1)

### Animation Time

Defines the coefficient animation time in milliseconds to make the dynamic animation flow smoothly.

### Animation Behaviour

Defines the animation's behaviour according to these two options:

- **Absolute:** the variable's maximum range value represents the set animation's end result using the measure unit that is compatible with the animation it refers to.
- **Proportional:** the variable's maximum range value is proportional to the animation's end value.

### Value Range

This property is only available with proportional type animations. It sets value range allowed for managing the animation:

- **Tag Min Value**
- **Tag Max Value**

### Repeatable

This enables animation repetition. When active, the animation will repeat cyclically for the entire time this option remains active.

### Vertical Movement

Defines the object's movement value on the Y axis in pixels, in respect to its starting point.

## 2.34.16. Composed Movement Animation

The Dynamic "**Composed Movement**" animation is used to determine animated movement along a trajectory which can be composed as pleased by using a "Polyline", "Polygon" or "Line" object. The trajectory can be defined as needed on screen by graphically drawing a path. It can also be set with simultaneous X and Y movements to create diagonal movements for example.





The use of this animation feature requires the user to insert a "Polyline", "Polygon" or "Line" element on screen from the Toolbox. Each point of the inserted polygon builds a path for the object or symbol to follow. The 'name' assigned to the "Polyline", "Polygon" or "Line" objects must therefore be specified in the dynamic movement function.

The "Polyline", "Polygon" or "Line" used as the path can be made visible during runtime by enabling the corresponding animation property provided for this purpose. Other elements from the 'Draws' category (eg. rectangle, Ellipse) used for defining the trajectory will have no effect on the symbol's animation which will remain in its position without moving during runtime.

## Composed Movement Animation Properties

### Animation Tag

This sets the tag to associate to the animation.

### Expression

This is used to set the expression whose result will be used in the animation.



Certain syntax can be used within this property for the following scopes:

- to connect the nth Bit of a Tag, insert the '.nBit' value.

(e.g. .1 to indicate Bit 1)

- to connect the nth element of an array, insert the '[nElem]' value.

(e.g. [1] to indicate the item 1)

### Animation Time

This defines the time coefficient, in milliseconds, in managing animation by making the dynamic animation smoother.

### Animation Behaviour

Defines the animation behaviour based on these options:

- **Absolute:** The variable's max. value range represents the final result of the set animation expressed with the measure unit compatible with the animation to which it refers.
- **Proportional:** The variable's max. value range is proportional to the animation's final value.

### Value Range

Sets the value range allowed for managing the animation.

- **Tag Min Value**
- **Tag Max Value**

### Repeatable

This is used for enabling/disabling animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active.

#### Path Control

Defines the name of the "Poly-Line" to be used as the trajectory path for the object to follow. Therefore, you will need to have already inserted a "Poly-Line" from the toolbox with set trajectory path points for the object to follow as well as a name so that it can be entered here.

#### Hide Path

This is used for hiding the trajectory path to be used by the polyline during project runtime.

#### Use Barycenter

This is used to indicate whether the movement reference point, in respect to the object, must be related to the object's centre.

### 2.34.17. Clipping Animation

The dynamic "Clipping" animation command is used to selectively display parts of the symbol according to the parameters set for this animation type.

## Clipping Animation Properties

#### Animation Tag

This sets the tag to associate to the animation.

#### Expression

This is used to set the expression whose result will be used in the animation.



Certain syntax can be used within this property for the following scopes:

- to connect the nth Bit of a Tag, insert the '.nBit' value.

(e.g. .1 to indicate Bit 1)

- to connect the nth element of an array, insert the '[nElem]' value.

(e.g. [1] to indicate the item 1)

#### Animation Time

This defines the time coefficient, in milliseconds, in managing animation by making the dynamic animation smoother.

#### Animation Behaviour

Defines the animation behaviour based on these options:

- **Absolute:** The variable's max. value range represents the final result of the set animation expressed with the measure unit compatible with the animation to which it refers.
- **Proportional:** The variable's max. value range is proportional to the animation's final value.

### Value Range

Sets the value range allowed for managing the animation.

- **Tag Min Value**
- **Tag Max Value**

### Clipping Type

Sets the clipping animation type to be applied.

## 2.34.18. Dynamic Borderline Animation

The Dynamic "**Border Line**" Animation command is used for assigning a variable to determine a dynamic dash stroke line with animated movement to frame objects such as the line or polyline objects, according to the following parameters.

Please take into account that the dashed borderline movement is mainly determined by the combination of three parameters: "Stroke Dash Offset", "Stroke Dash Cap" and "Animation Time". The velocity will be determined by the fact that the distance set will be covered in the Animation Time set. In principle, the following formula is to be considered to determine the velocity of the dash line movement:

Stroke Dash Offset" – "Stroke Dash Cap"

## Dynamic Border Animation Properties

### Animation Tag

This sets the tag to associate to the animation.

### Expression

This is used to set the expression whose result will be used in the animation.



Certain syntax can be used within this property for the following scopes:

- to connect the nth Bit of a Tag, insert the '.nBit' value.

(e.g. .1 to indicate Bit 1)

- to connect the nth element of an array, insert the '[nElem]' value.

(e.g. [1] to indicate the item 1)

### Stroke Thickness

Defines the thickness of the border line in pixels.

### Stroke Dash Array

Used to set the line type to be displayed when animation is active. In addition to the options proposed by the drop down menu, the user can also set the displayed line type using an array of values (eg. 1,2,1,3 ecc.). In this case, the first value which corresponds to the 0 index, specifies the dash length, the second value which corresponds to the 1 index, specifies the gap length between the dashes, the third, if present, will specify the dash length again and so on.

### Stroke Dash Offset

Defines the distance that the animation trajectory has to cover in the time defined in the 'Animation Time' property. The borderline's scrolling time will be determined by these two parameters and the "Stroke Dash Cap". A negative value changes the direction of the rotation..

### Stroke Dash Cap

Defines the animation's Stroke Dash Cap value. Basically, this value will be subtracted from the "Stroke Dash Offset" parameter to determine the real distance that the animation should cover within the Animation Time set.

### Animation Time

Used to set the coefficient time (in milliseconds) in the animation's management to make the dynamic animation flow more smoothly.

### Repeatable

This is used to enable animation repetition. When enabled the animation will be repeated in a loop while Animation Condition remains active.

### Auto-Reverse

This is used to enable the auto-reverse making the same animation flow backwards when executed.

## 2.34.19. Dynamic Text Animation

The Dynamic **"Text"** command is used for activating the "Change Text" function in the selected object. This will allow the component to change the text displayed in runtime in function with the associated variable value.

### Animation Tag

This sets the tag to associate to the animation.

### Expression

This is used to set the expression whose result will be used in the animation.



Certain syntax can be used within this property for the following scopes:

- to connect the nth Bit of a Tag, insert the '.nBit' value.

(e.g. .1 to indicate Bit 1)

- to connect the nth element of an array, insert the '[nElem]' value.

(e.g. [1] to indicate the item 1)

## Dynamic Text Animation Properties

### Format

This is used to set the text string format.

## 2.35. Command Explorer Window

### 2.35.1. Command Settings

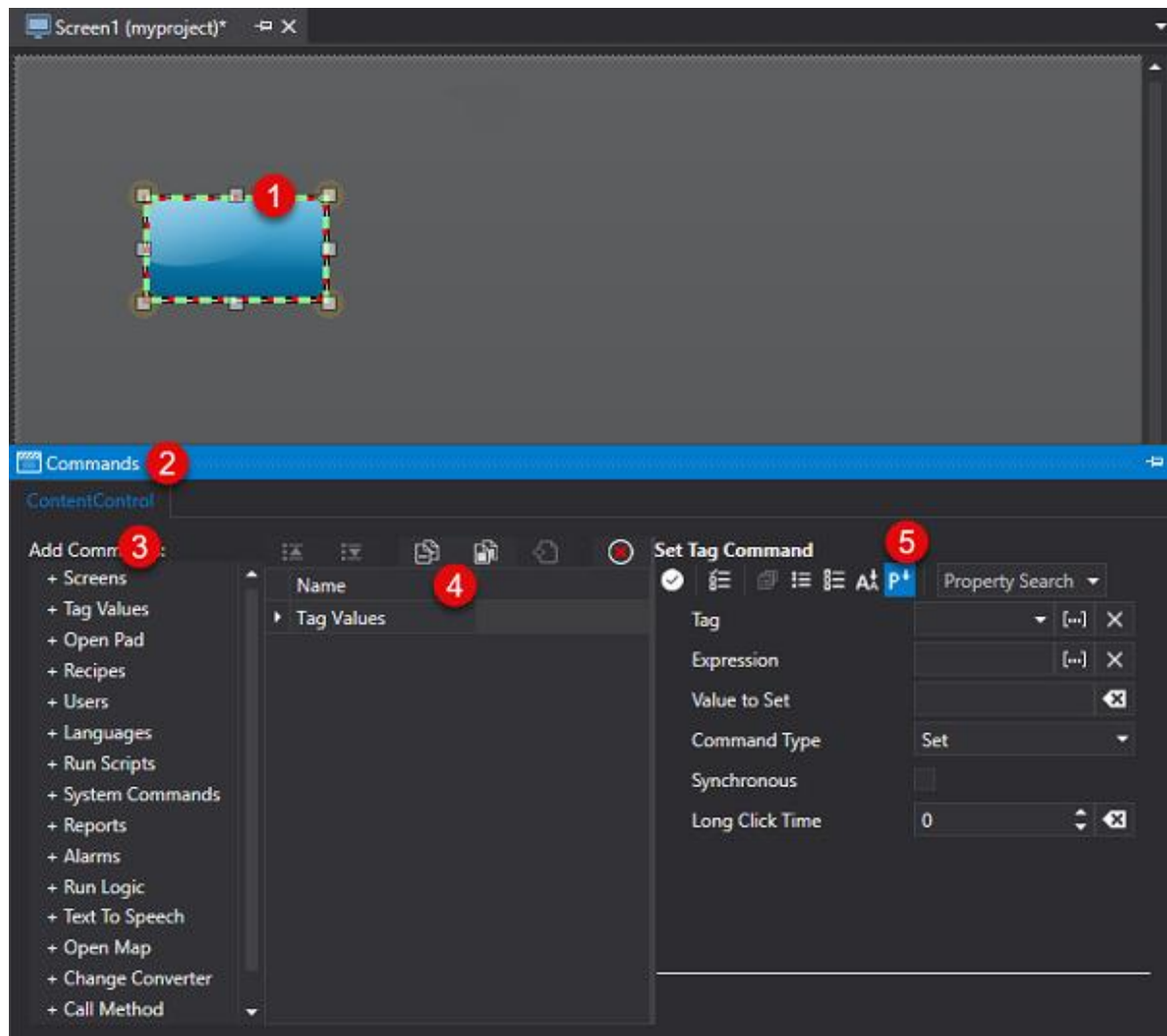
The commands in Movicon.NExT Objects and Symbols, and the association of Tags that determine the execution of a command by clicking the object, on a click, can be set using the "Command Explorer".




The name of the reference Tag for each command inserted in the object can be specified in the "Tag" property.  
In cases where the Tag has not been specified for each individual command, the object will use the reference Tag of the one associated to the object (see paragraph: "Assigning Tags to Objects").

### Configuring commands

To configure a command of an object proceed as follows:



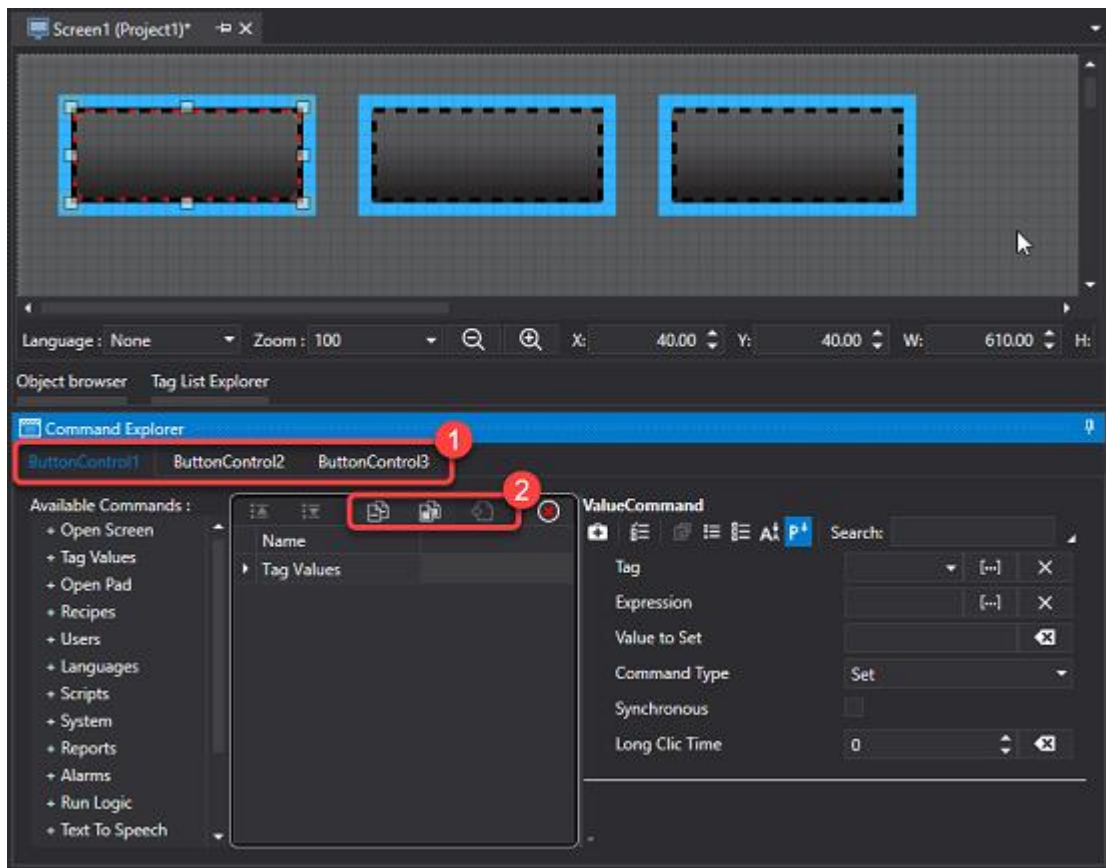
1. Open a screen and select the desired graphical object.
2. Call the "Command Explorer" window.
3. In the left part of the window you will find a list of commands that can be inserted. Click on the name of the command you wish to insert for the object.
4. The command will then be added to the list ready for configuring.
5. After having added the desired command, you can now proceed with configuring it by means of its properties.

 More than one command can be added to each individual object.

### Configuring commands in several objects

When multi-selecting a number of objects, you will be able to insert the desired commands in each one.

1. The same number of tabs as there are selected objects (each with the object's name) will show in the "Command Explorer" window. When selecting one to the tabs, the corresponding object's Command List will activate allowing you to edit it.
2. You can copy commands from one object to another by using the Copy, Copy All and Paste buttons.



### 2.35.2. Commands for Screens

The "Screens" command set to execute operations in project screens can be configured with the following properties:

#### Screen Name

Name of screen to be opened or closed.

#### Monitor ID

Next supports the multi-monitor management. The "Monitor ID" parameter is managed to identify where the screen it is to be displayed when opened according to this parameter's settings. The requested monitor can also be indicated in the screen's properties. However, the 'Open Screen' command button has priority over the Screen parameters if both have been set.



The "-1" value identifies the default monitor set in Windows. When opening a screen in a different monitor that is not the one set for default (therefore by setting a value that is different from -1), it will always open at the "0;0" coordinates. This will also happen with the "PopUp" and "Shared" commands and when specifying the open coordinates.

Monitor identification can be obtained from the operating system's menu which identifies connected monitors.

#### Command Type

The options are:

- **Open Normal:** opens the screen specified in the "Screen Name" property in normal mode as classic change screen. When the "Screen Name" property is left blank, the command will load the previous screen, by executing the "Close and return to previous screen" command.
- **Open Modal Popup:** opens the screen specified in the "Screen Name" property as a modal window.
- **Open Non Modal Popup:** opens the screen specified in the "Screen Name" property as a non modal window.
- **Open Context Popup:** opens a contextual popup window.
- **Open Shared:** This screen opening mode is no longer supported and screens opened in this mode will open in Normal mode.
- **Close:** when the "Screen Name" property is left empty, this command will open the "Tile Page" window or the startup Screen according to the project's startup mode. When a name of a screen is inserted in the "Screen Name" property and opened as "Synchro", the modal screen will close. When the screen is opened as "Normal" mode, a Back command will be executed to load the previous screen.

### Parameter File

This edit box is selected using the "..." browse button on the right of the name, and any eventual path, of the parameter file when needing to open a screen by passing it parameters.

For further information on using screen parameterization, please see the paragraph on: "Screen Parametrization".

### Inherit Parameter File

When enabled this property will make the parameter file inheritable when multiple screens are opened in cascade mode. Basically, from within one screen opened passing it the parameter file will open another with its Inherit Parameter File property enabled and this second screen will use the same parameter file passed to the first screen opened.

### Use Relative Position

This parameter is considered only when the "Shared" mode has been selected. In this case the screen, whether Modal or Frame, will open at the X and Y coordinates inserted in the appropriate fields in absolute mode (False value) or in the relating called screen mode (True value). When set to True, the screen, whether it be Modal or Frame, will open at the coordinates indicated by taking the called screen's position as origin and not the monitor screen's origin.



This parameter is only considered when the open screen command is invoked from a Screen's object, such as a button. It will not be considered if the command is invoked from a resource such as a Menu, Accelerator or an Event resource and other such like.

### X Position

This parameter is only considered when "Shared" mode has been selected. In this case the screen, whether Modal, Frame or Syncro will open in the X position indicated by the parameter and not in the centre of the monitor.

### Y Position

This parameter is only considered when "Shared" mode has been selected. In this case the screen, whether Modal, Frame or Syncro will open in the Y position indicated by the parameter and not in the centre of the monitor.

### AutoClose Secs



This parameter is only considered when the "Shared" screen opening mode has been selected. When inserting a value that is different from zero in this field, the screen, whether Modal or Frame, will close automatically when time expires (expressed in seconds). For example, if you insert the value 5, once open the Modal or Frame screen will automatically close after 5 seconds.

### Calling Screen Offset

When this is enabled in multi-monitor applications, it will allow the pop-up window to open in relation to the screen calling it. The set coordinates will be given reference to the monitor where the calling screen is present.

### Long Click Time

The command will be executed normally with the default value. However when using a value that is higher than 0, the command will be executed only when pressed down for the duration set in seconds. The commands executed in this mode (delay) exclude any normal commands eventually associated and vice versa.



This mode is not available in **webclient**; when keeping a button pressed down, all its associated delay commands will be executed once only in sequence.

## 2.35.3. Commands for Tag Variables

The "value" command type settings are used for setting value associations to project Tag variables. This command's parameters are defined by configuring the following properties:

### Tag

When the command is associated to an object from the toolbox, such as a Button or Hot Region, and this Tag filed is left empty, the command will be applied to any default variable associated to the object.

### Value to Set

Value which will be set in the variable or which will be used as increase/decrease step.



Attention! In cases where a decimal value is inserted (eg. 1.3 etc.) you will need to use the point (".") as a decimal separator, independently of Windows settings.

### Command Type

Possible command types include:

- **Set:** sets the variable with the value defined in the "Value" parameter
- **Increase:** increases the variable of the value defined in the "Value to Set" parameter.
- **Decrease:** decreases the variable of the value defined in the "Value to Set" parameter.
- **Toggle:** toggles the variable between the "0" value and the one set in the "Value" parameter
- **Impulsive:** when pressing this button, the variable will set to the value defined set in the "Value to Set" parameter and will remain set until button is released. This will happen when setting the "Impulsive Time" property with the "0" value. If the "Impulsive Time" parameter is set at with a higher value than zero, this value will be considered as the timeout for resetting the variable. In this case, when the button is pressed, the variable will be set with the value in the "Value to Set" parameter and will be reset to zero at the release of the button or when time expires in cases where button is still pressed.



This command is not supported by WebClient (WPF/HTML5), but is supported by WebClient (SVG/HTML5).

- **Impulsive Latch:** pressing this button, the variable will set to the value defined in the "Value to Set" parameter and will be kept set with this value until the time set in the "Impulsive Latch" parameter has expired, independently from the fact that the button is kept pressed or released. In cases in which the "Impulsive Latch" parameter is set at zero, the command will behave like the "Impulsive" command, that is to say that the value in the variable will be maintained for all the time in which the button remains pressed.



This command is not supported by WebClient (WPF/HTML5), but is supported by WebClient (SVG/HTML5).

- **TransferValue:** This allows the value of one variable to be transferred to another variable. In this case the original value is the one specified in the 'Tag' parameter and the destination variable is the one specified in the 'Destination Tag' parameter. Different variable types can also be used whereby Movicon.NExT will perform the appropriate conversions providing that the two tag types are compatible to allow the correct value transfer. For example it will not be possible to transfer alphanumeric text of one variable string to a numeric variable.



The "Transfer Value" command always updates the whole Tag. This means that if an Array element or a Tag Bit is defined in the Expression field, the whole destination Tag will be however written and therefore all the Array elements and Tag bits will be updated. Basically, in this context it would not make sense to specify the element of an Array or the Bit of a Tag.

- **NumericPad:** This command opens a Numeric pad as an alternative to using the keyboard for inserting values. The value will be reported to the variable once confirming with the 'OK' pad key. This numeric Pad is fundamentally used with Touch-screens.



**Numeric Variables:** when the variable is a numeric type, the number set will be represented by the same number of decimal figures after the point. For example, the '2' value will be set in the 'x.xx' format. When the set value is negative, the absolute value will be taken, therefore when entering '2' or '-2' the same result will be achieved. When the value is a floating point type with decimal figures, the value will be rounded off to the nearest number. Therefore when entering '1.4', it will be rounded off and entered as '1' and when entering '1.6,' it will be rounded off and entered as '2'.

- **AlphanumericPad:** This command opens a Alphanumeric Pad as an alternative to using the keyboard for inserting characters. The value will be reported to the variable once confirming with the 'OK' pad key. This numeric Pad is fundamentally used with Touch-screens.
- **ResetStatistic :** Resets the statistics information, when enabled, of the variable associated to the command.
- **Append Value:** appends the "Value to Set" the actual Tag performing a chain of strings even when the value is numeric.
- **Remove Value:** removes the last character on the right of the Tag's value, even when the value is numeric.

- **Change Sign:** this command allows the Tag's value to be changed from positive to negative or vice-versa. This command is toggle type, meaning that each time it is executed, it will invert the tag's sign.
- **Append ON-OFF Decimal:** this command allows to define whether the "Append Value" command or "Remove Value" should be performed on the entire tag or only on its decimal part. This command is only valid for floating point tag types. This command is toggle type and therefore, each time it is executed, it will invert the entry position in respect to the current one.

### Synchronous

When this option is enabled, the command will be executed synchronized with the User Interface. This means that if an open screen command comes after the Tag's setting command in the command list, the screen will only open after the Tag has been set,

### TransferToTag

This field shows only when the 'TransferValue' command has been selected. This field is used to select the destination Tag to which to transfer the tag value, defined in the 'Tag' field.

### Long Click Time

This field is not available for "Impulsive" and "Timed Impulsive" command types.

When setting the major value 0, the command will only execute after having been kept pressed for the time set in seconds.

When set with the default value, the command will execute normally.

The commands executed in this way (delay) exclude any commands they have been associated with and viceversa.



This modality is not available in **webclient**; when a button is kept pressed down, all the delay commands that it has been associated with will be executed once only in sequence.

### Expression

This is used to set the expression associated to the command.

This will make it possible, for example, to execute the command on a specific Tag bit or on a specific element in cases where the Tag is an array type. It is also possible to insert an expression for the 'Transfer Value' command only, to determine the destination value (eg.  $= [x] + 10$ ).

### TransferToTag

This field will only appear when the 'TransferValue' command type has been selected and it is used to select the destination Tag in which the variable value defined in the 'Tag' field will be transferred.

### MinValue

This field only appears when the "NumericPad" or "AlphanumericPad" Command Type has been selected and it is used to define the minimum value or minimum number of characters that the Tag can be set with.

When setting the "MinValue" value equal to the "MaxValue" value, the Pad will open without imposing limits on the value to be set.

### MaxValue

This field appears only when the "NumericPad" or "AlphanumericPad" Command Type has been selected and it is used to define the maximum value or maximum number of characters that Tag can be set with.

When setting the "MinValue" value equal to the "MaxValue" value, the Pad will open without imposing limits on the value to be set.

### TagMinValue

This field only appears when the "NumericPad" or "AlphanumericPad" Command Type has been selected and it is used to select a project Tag whose value will be taken in runtime as the minimum value or minimum number of characters that the Tag can be set with. In this way, this value can be made dynamic in runtime.

When setting the "MinValue" value equal to the "MaxValue" value, the Pad will open without imposing limits on the value to be set.

### TagMaxValue

This field only appears when the "NumericPad" or "AlphanumericPad" Command Type has been selected and it is used to select a project Tag whose value will be taken in runtime as the maximum value or maximum number of characters that the Tag can be set with. In this way, this value can be made dynamic in runtime.

When setting the "MinValue" value equal to the "MaxValue" value, the Pad will open without imposing limits on the value to be set.

### Decimals

This field only appears when the "NumericPad" Command Type has been selected and it is used to define the number of decimal figures to be displayed. When associating the Numeric Pad with Floating type variables, this parameter determines the number of decimal figures displayed. When the integer type variables are associated to the Numeric Pad, this parameter allows the value to be displayed with a floating point. For example, when setting the field to 2 with the integer variable with a 123 value, the value will display as 1.23. In this case, the minimum and maximum limits will not take into account the set format and will continue to work in the variable's entire value.

For example, with a one decimal figure format and a maximum limit of '100', it will be possible to edit a maximum value in the variable equal to '100' and therefore the value displayed in the Numeric Pad will be a maximum of '10.0'.

### Ignore Decimal

This property is only available for "Append Value" and "Remove Value" type commands. When enabled, it allows the decimal point to be ignored of the value to be edited. In this way it will be possible to also append or remove the "." or "," characters from the string by considering them like all the other characters. When this property is left disabled, any "." or "," characters (according to the international Windows settings) inserted in the string will be considered as decimal separators.

### X Position

This is used to set the X position in pixels where the Pad window will open. The window will open in the same monitor of the screen calling it. The -1 value opens the window in the center of the screen.

### Y Position

This is used to set the Y position in pixels where the Pad window will open. The window will open in the same monitor of the screen calling it. The -1 value opens the window in the center of the screen.

### Use Relative Position

This is used to decide whether the coordinates to be considered are relative or absolute to the calling screen. This option is considered only when the "X Position" and "Y Position" parameters are different from the default value (Pad opening in the center of the screen).

### 2.35.4. Open Pad Commands

This command type can be reached through the "Command Type" properties for commands in variables. It is not supported in HTML WebClient or in Apps.

#### Tag

This field is used for selecting the Tag in which the selected Command Type will be executed. When the command is associated to an object from the toolbox, such as a Button or Hot Region or other symbols, leaving this Tag field blank will apply the command to any contextual variable associated to the object.

#### Expression

This is used to set the expression associated to the command. For example, the command can be executed in a specified Tag bit or in a specific element if the Tag should be an array type. It is only possible to enter an expression for the "Transfer Value" command to determine target value (e.g.. = [x] + 10).

#### Reverse Expression

This is used to set the Reverse Expression associated to the command.

#### Command Type

This is the command type that can be used. The choices are:

- **Numeric Pad:** this command executes the opening of a Numeric Pad through which values can be entered without using the keyboard. The value will be reported to the variable after confirming with the "OK" key. This is usually used with Touch Screens.



When Floating Point type variables are associated to the Pad, this parameter determine number of decimal digits to be displayed. When integer variable types are associated to the Pad, this parameter will allow the value to be displayed with a dummy point. For example, when setting this field to 2, the integer variable with the 123 value will be displayed in the Display as 1.23. In this case, the minimum and maximum limits and the spin pace will not take into account the format set in the display and will always work with the variable's integer value. For example, with a decimal digit format and a "100" maximum limit, it will possible to edit a maximum value equal to "100" so that the value shown in the Display will be a maximum of "10.0".

- **Alphanumeric Pad:** this command executes the opening of an Alphanumeric Pad through which it characters can be entered without using the keyboard. The value will be reported to the variable after confirming with the 'OK' key. The use of this pad is essential for touch screens.

#### Synchronous

Enabling this option will synchronize the command with the User Interface. This means that if there is a command to open a screen in the command list after the Tag's command settings, the screen will only be opened after the Tag has been set.

#### Minimum Value

This field only appears when the "NumericPad" or has been selected, and it is used to set the minimum value or minimum number of characters that can be set in the Tag. When setting the 'Minimum Value' value to the same value of the 'Maximum Value', the Pad will open without any limits as to which value can be set.

### Maximum Value

This field appears for both the "NumericPad" and "AlphanumericPad" command types when selected, and it is used to define the maximum value, or the maximum number of characters, that can be set in the Tag. When setting the 'Minimum Value' value to the same value of the 'Maximum Value', the Pad will open without any limits as to which value can be set.

### Minimum Value Tag

This field only appears when the "NumericPad" command type has been selected, and it is used to select a Tag from the project whose value will be taken as the minimum value or minimum number of characters that can be set in the Tag at runtime. This will make this value dynamic at runtime.

When setting the value of the 'Minimum Value Tag' to the same value of the 'Maximum Value Tag', the Pad will open without any limits as to which value can be set.

### Maximum Value Tag

This field appears for both the "NumericPad" and "AlphanumericPad" command types when selected, and it is used to select a tag from the project whose value will be taken as the maximum value or maximum number of characters that can be set in the Tag at runtime. This will make this value dynamic at runtime. When setting the value of the 'Minimum Value Tag' to the same value of the 'Maximum Value Tag', the Pad will open without any limits as to which value can be set.

### Number of Decimals

This field appears only when selecting the "NumericPad" command type and is used to define the number of decimal digits to display. When Floating Point variable types are associated to the Numeric Pad, this parameter will determine the number of decimal digits are to be displayed. When integer variable types are associated to the Numeric Pad, it allows the value to be displayed with a dummy point. For example, When the integer variable is 123 and you set this field to 2, the value will be shown as 1.23 in the display. In this case the minimum and maximum limits will not take into consideration the set format and will always work with variable's value as a whole. For example, with a one decimal digit and a maximum limit of '100', it will be possible to set the maximum variable value to a '100' and therefore the value displayed in the Numeric Pad will be a maximum of '10.0'

### Value in Password Style

This sets the value to display in password mode where asterisks will replace the digits to hide them from view.

### Long Click Time

This field is not available for the "Impulsive" and "Time Impulsive" command types.

When setting a value higher than 0, the command will only be executed after being pressed for the time set in seconds.

When set with the default value, the command will be executed as normal.

The commands executed in this mode (delayed) exclude any normal commands and viceversa.



This modality is not available with **Web client**; Keeping the button pressed will execute all the delayed commands, it has been associated with, once only in sequence.

### Enable Engineering Unit

Enabling this (True for default) will allow the pad to take into account of the engineering unit limits of the tag associated to the command when opened on screen. When disabled, it will take into account the any static or dynamic limits set for the command.

### X Position

This is used to set the X position in pixels where the Pad window will open. The window will open in the same monitor of the screen calling it. The -1 value opens the window in the center of the screen.

### Y Position

This is used to set the Y position in pixels where the Pad window will open. The window will open in the same monitor of the screen calling it. The -1 value opens the window in the center of the screen.

### Use Relative Position

This is used to decide whether the coordinates to be considered are relative or absolute to the calling screen. This option is considered only when the "X Position" and "Y Position" parameters are different from the default value (Pad opening in center of screen).

## 2.35.5. Recipe Commands

The properties below are used for configuring the parameters of command executions of Recipes inserted in the project:



This assumes that the recipe commands refer to a local client and server. If you are using a remote client, you will need to specify the connection session for the recipe and enter the name of the session in the client's settings and specify the server that the recipe must connect to.

### RecipeName

Name of Recipe to be called.

### CommandType

The command type that can be executed for a recipe. The options are:

- **Show:** opens the pop-up window with the Recipe defined in the "Recipe Name" parameter
- **Load From DB:** loads the values of the Recipe defined in the "Tag Recipe Index" from the DataBase and transfers them to the "Tag Data Value" defined for the recipe fields (see topic on "Using Tags to manage recipes")
- **Save To DB:** saves the values set in the "Tag Data Value" defined for the recipe field in the DataBase with the Recipe name defined in the "Tag Recipe Index". If the recipe does not exist in the DataBase, a new record will be added. If it already exists, it will be modified instead. (see topic on "Using Tags to manage recipes").
- **Delete From DB:** removes the recipe with name defined in the "Tag Recipe Index" from the DataBase (see topic on "Using Tags to manage recipes").
- **Write to PLC:** activates the Recipe with the name specified in the "Tag Recipe Index" (see topic on "Using Tags to manage recipes"). The values will be loaded from the DataBase and transferred to the device. If a recipe does not exist with this name, an error will be issued in the Tag Recipe State. Attention: the values that are transferred to the device are read by the DataBase and not by the recipe's Tag Data Value". In this case, a recipe is loaded and then modified using the "Tag Data Values". If the recipe is activated before being saved with the "Save Recipe" command, the old values in the database will be transferred. This command can nevertheless be executed without defining the recipe's "Tag Data Values".
- **Read From PLC:** reads the device values and copies them to the recipe's "Tag Data Value" (see topic on "Using Tags to manage recipes").

- **Export/Import from file:** Exports/imports the selected recipe to/from the File.

### Synchro

Waits for command execution before consenting user to perform operations.

### Command TimeOut

TimeOut for executing the "Activate" and "Read" recipe commands. This time is set in milliseconds.

### Delay Command Secs

This command functions normally when set with the default value. When set to a value higher than 0, it only functions when kept pressed down for duration of the time set in seconds.

Commands activated in this mode (delay) exclude any normal commands eventually associated and viceversa.



This modality is not available in **webclient**; when a button is kept pressed down, all the delay commands that it has been associated with will be executed once only in sequence.

## 2.35.6. User Commands

The parameter settings of "Users" command types set to execute operative commands for Users in the Password and Protection management are configured in the following properties:

### CommandType

The command type to be executed.

- **Login:** executes user login. A dialog window opens for inserting the user's name and password.
- **Logout:** executes logoff of the user currently logged in.
- **UnlockUser:** A string type tag can be associated to the command with the name of the user to be re-enabled as its value. In cases where a tag is not specified or has a null value, a new popup window will open containing a list of users names that have been blocked within the project.
- **UnlockAllUsers:** This command unlocks all the users disabled within the project.
- **Edit:** Opens the users edit window through which it is possible to add or change users at runtime.
- **Change password:** Sets the command to change password using a pop-up that displays after activating the command.



To set the desired restrictions for the "edit" command, you will need to specify the "access level" or "access role" properties of the button used to invoke the command.

### Delay Command Secs

This command functions normally when set with the default value. When set to a value higher than 0, it only functions when kept pressed down for duration of the time set in seconds.

Commands activated in this mode (delay) exclude any normal commands eventually associated and vice versa.





This modality is not available in **webclient**; when a button is kept pressed down, all the delay commands that it has been associated with will be executed once only in sequence.

### 2.35.7. Change Language Command

The "Change Culture" command settings are used for establishing Change Language executions for project text strings using the languages (Cultures) defined in the String Table. This command's parameters can be defined by configuring the following properties:

#### Culture Name

This field is used for selecting the language to activate from a list of language names. This list is populated with the languages that have been inserted in the project's String Table.

If this field is left empty, Movicon will display a window through which the user can choose a language to activate from those available in Runtime.

#### Delay Command Secs

This command functions normally when set with the default value. When set to a value higher than 0, it only functions when kept pressed down for duration of the time set in seconds.

Commands activated in this mode (delay) exclude any normal commands eventually associated and viceversa.



This modality is not available in **webclient**; when a button is kept pressed down, all the delay commands that it has been associated with will be executed once only in sequence.

### 2.35.8. Script Execution Commands

The following properties are used to define the parameters of a "Script" command to execute a previously inserted VB.NET Basic Script resource in the project:

#### ScriptName

The name of the script to be called.

#### Parameter

This field is used for inserting a string which will be passed to the script as its execution parameter. This parameter can be retrieved in the script's code with this function:

```
StartupContext.Parameter
```

When dealing with a string, it can be formatted in way to provide its passage through several parameters separated by a predefined character. For example, a series of values can be passed by separating them with the "pipe" character:

```
1|5,6|29
```

In this way once the parameter has been retrieved, the string can be split using the "pipe" character as a separator.

### ExecutionMode

The script execution mode:

- **Syncro**: script execution is synchronized with call
- **Normal**: default mode, script execution is asynchronized with call
- **Shared**: The script shares the same execution thread of the other scripts that have been started up in the same way. This consents a more limited use of the machine's resources
- **Stop**: this command stops the script from running and loads it into memory. In this way variable subscriptions used by the script will be removed from the Server according to the time set in the "Connection Settings" (belonging to the project or the script itself if this has its own session).

### Tag

Sets the Tag associated to the command.

### Express

Sets the expression associated to the command.

### Delay Command Secs

This command functions normally when set with the default value. When set to a value higher than 0, it only functions when kept pressed down for duration of the time set in seconds.

Commands activated in this mode (delay) exclude any normal commands eventually associated and vice versa.



This modality is not available in **webclient**; when a button is kept pressed down, all the delay commands that it has been associated with will be executed once only in sequence.

## 2.35.9. System Commands

The following properties are used for configuring the settings of System Commands to execute a command type selected from the proposed list of commands:

### CommandType

The type of command to be executed.

- **Quit Runtime Execution**: Stops the project being run.
- **Computer Logoff**: stops the project being run and logs off the Windows user.
- **Lock Computer**: executes the Windows' Lock command.
- **Shutdown computer**: stops the project being run followed by Windows shut down.
- **Restart Computer**: stops the project being run followed by Windows restart.
- **Run Application**: runs or stops the application specified in the "Command Parameters" field based on the "Execution Mode" field.

### Parameter

The name and path of the application name to be started up or stopped using the **"RunApp"** command is specified in this field. The path can be omitted if the application has been inserted in variables within the Windows (e.g. C:\Windows\system32\notepad.exe, or Notepad.exe)

### ExecutionMode

The **"RunApp"** command's execution modes. The mode options are:

- **Normal:** the application indicated in the **"Parameter"** field will be started up normally i.e. not synchronized with Movicon.
- **Syncro:** the application indicated in the **"Parameter"** field will be started up in synchronized mode with Movicon, i.e. the Movicon interface will not be accessible until the application has terminated.
- **Shared:** not used.
- **Stop:** the application indicated in the **"Parameter"** field will be terminated.

### Delay Command Secs

This command functions normally when set with the default value. When set to a value higher than 0, it only functions when kept pressed down for duration of the time set in seconds.

Commands activated in this mode (delay) exclude any normal commands that may have been associated and viceversa.



This modality is not available in **webclient**; when a button is kept pressed down, all the delay commands that it has been associated with will be executed once only in sequence.

## 2.35.10. Commands for Reports

"Reports" command types can be set and used for Reports inserted in the project. The Report command type settings are configured using the properties below:

### Report Name

Name of the report to be displayed or managed.

### Command Type

Commands can be display or print type.

- **Show:** opens the report
- **Print:** prints the report specified in the "ReportName" property using the system's default printer
- **PrintDialog:** prints the report specified in the "ReportName" property by first opening the window to select the printer to be used
- **Send:** Send the report specified via mail, using the SMTP Server defined into General User Management Settings
- **Save:** Saves the specified report

### Report Opening Mode

This is used to set the mode with which to open Reports:

- Normal: Full screen
- Syncro: Modale pop-up window

- Shared: Non-modal pop-up window
- Stop: Close report

#### File Name

Sets the file name with which the report will be saved.

#### File Type

Sets the file type with which the report will be saved/exported.

#### Recipient

This field is used to enter the name of the user or user group to send the email with attached report file to.

#### From

Used to set the sender's address to be displayed in the email.

#### FromAlias

This is used for setting a "alias" name which will display in the email instead of the sender's address.

#### Mail Subject

Sets the email's subject.

#### Mail Object

Sets the email's text (body)

#### Parameters

This is used to set the list of parameters to pass to the report when opened.

#### Connection String

This field is used for configuring the connection string towards a database which is different to the one defined in the Report properties. When left empty, the Report will use the one set in its properties.

#### Data Load Timeout

This timeout is used for performing data extraction queries. When leaving this property set with the zero value, the provider's default time will be used which is usually 30 seconds.

#### Delay Command Secs

This command functions normally when set with the default value. When set to a value higher than 0, it will only function when kept pressed down for the duration of the time set in seconds.

Commands activated in this mode (delay) exclude any normal commands eventually associated and vice versa.



This modality is not available in **webclient**; when a button is kept pressed down, all the delay commands that it has been associated with will be executed once only in sequence.

### 2.35.11. Alarm Commands

This group includes commands to acknowledge and reset Movicon alarms.

## General

### Type

Sets the command type required

**AckAll:** This command is used to acknowledge all active alarms.

**ConfirmAll:** This command confirms all the active alarms by resetting them.

**ToggleSound:** This command enables or disables the warning sound that is activated for those alarms which have not yet been acknowledged according to which user is active.

**ShowStatisticReport:** This command enables a preview of the selected report to be shown in DevExpress format for Alarm statistics.

**PrintStatisticReport:** This command prints the selected report of the alarm statistics in DevExpress format without opening a print preview window beforehand.

**SaveStatisticReport:** This command creates and saves a new report file in the format selected from the FileType property. For further information please go to the 'Various' > 'FileType' section.

**SendStatisticReport:** This command exports and sends the selected alarm statistics report by email. The email is sent in accordance with the properties set in the SMTP settings in the "Users and Groups".

The email recipient will be the user or user group defined in the property.

### ReportType

The three types of default Reports interface with the project's Historian Log and show the following information:

- OrderByDate: Report showing alarms in date order
- OrderByDuration: Report showing alarms in order of duration (starting with the longest to the shortest duration).
- OrderByOccurrence: Report showing alarms grouped by occurrence.

### Recipient

The user or user group, to send the email with attached report file to, is entered in this field.

This parameter is only shown when the "**SendStatisticReport**" command has been selected from the "Type" field.

### DB ConnectionString

This field is used for configuring the connection string towards the DB server in which the referenced Historian log database resides.

### Data Load Timeout

This timeout is used for performing data extraction queries. When leaving this property set with the zero value, the provider's default time will be used which is usually 30 seconds. This property is available for **Show Statistics Report**, **Save Statistics Report**, **Print Statistics Report** and **Send Statistics Report** command types.

### Long Click Time

This command functions normally when set with the default value. When set to a value higher than 0, it only functions when kept pressed down for the duration of the time set in seconds.

Commands activated in this mode (delay) exclude any normal commands that may have been associated and vice-versa.



This modality is not available in **webclient**. When this command button is kept pressed down, all the associated delay commands will execute once in sequence only.

## Various



Each of the following properties display according to which command has been selected in the previous "Type" section.

### PeriodType

Used for setting the referential period of time for collecting data in the report.

### FileName

Used for setting the file name with which the report will be saved.

### FileType

Sets the file type with which the report will be saved/exported.

### From

Sets the sender's address to be shown within the email

### FromAlias

This property allows you to set an "alias" that will be shown instead of the senders name within the email

### MailSubject

Sets the email's subject.

### MailObject

Sets the email's body text.

## 2.35.12. Logic Commands

The settings of logic type commands are used to start or stop logic executions.

### Logic Name

This field is used to select the logic name.

### Execution Mode

- **Normal:** logic is started and kept in continuous cyclic execution. A 'Stop' command must be used to stop logic execution which will however stop when project stops running.
- **Synchro:** logic is executed once only then stopped automatically. Therefore only one complete cycle of the logic will be executed.
- **Shared:** logic is started up in "debug" mode, meaning that it will be started up in 'Normal' mode with an opened debug window showing the logic blocks edited in design mode. These

logic blocks will be animated in the debug window according to the Tag values and combined logic results. Closing the debug window will also stop logic execution.

- **Stop:** Stops the logic set in the command.

### Delay Command Secs

The command is executed in normal mode when set with the default value. When set with a value higher than 0, the command will only be executed when pressed down for the time set in seconds.

The commands executed in this mode (delay) exclude any normal commands associated and vice versa.



This mode is not available in the **webclient**; all the delayed commands associated to a button are executed once only in sequence when it is pressed down, .

### 2.35.13. Text to Speech Commands

The settings of "Text to Speech" command types are used to vocally synthesize a string inserted in the command. This feature requires the use of a speaker to reproduce the desired text string into speech.

The Text To Speech command can be used wherever possible to define a command list.



This feature is not available on **webclient**

### Voice Name

This is used to specify which voice to use among those installed in the system and displayed in the Text To Speech section in the control panel.

This field can also be left empty to use the voice selected for default in the control panel's Text To Speech section.

### Volume

The value range starts from 0 to 100 to indicate the volume intensity to use when reproducing the command into speech.

### Rate

Indicates the voice speed rate ranging from 0 being the slowest to 10 being the fastest.

### Speak

Represents the 'read' text string by the command execution. If the text is presented as string ID in the text table, the string of the language active at runtime will be used.

### Long Click Time

When set with the default value, this command will be executed normally. When set with a value higher than 0, this command will only execute when pressed for the time duration set in seconds.

Commands executed in this mode (delayed) will exclude any normal commands associated and vice versa.

### Execution Mode

This property can assume the following values:

**Normal:** executes asynchro speech

**Synchro:** executes speech and waits until terminated

**Shared:** cancels all pending speak and emits an asyncho speech

### 2.35.14. Open Map Commands

The Open Map command settings support normal mode to open maps in the mainframe or synchro mode to open modal geomaps.

The command also supports geographical areas that are zoomed upon opening the map.

This function can be used to invoke modal geomaps in the zone desired or to create a main map invocation layout screen in different areas.

#### Execution Mode

This is used to select the which mode to open the map with.

#### Zoom To

This is used to insert the coordinates with which the map will open up on.

#### Enable Zooming Scrolling

Activates the possibility to use the scroll and zoom on the map.

#### Show MiniMap

Activates the minimap at the top left.

#### ShowNextButton

Allows the Next button to be shown to navigate through the screens on the GeoPage.

### 2.35.15. Change Converter Commands

The Change Converter commands is used to change the converter type used in all those objects that have been set with the same converter.

For further information regarding the use of converters please refer to the topic on Unit\_Converter

#### Convertor Name

The name of the convertor is set here.

#### Execution Mode

- Normal: activates the selected convertor
- Stop: deactivates the selected convertor. When the active convertor is deactivated, the variable will only be displayed with its effective value.
- Syncro: not supported
- Shared: not supported

#### Delay Command Secs

This command is executed in normal mode when set with the default value. When setting this command with a value that is higher than 0, it will only execute when kept pressed down for the time set in seconds. Commands executed in delay mode exclude any normal commands associated and vice versa.



This mode is not available on the **webclient**; when keeping the button pressed down, all the delay commands associated to it are only executed once in sequence.



### 2.35.16. Commands in Methods

The settings of a "Call Method" command type sets the execution of a Method function. The following properties can be configured for setting this command:

#### Tag

This field is used for selecting the name of the method to be called.

#### Delay Command Secs

This command functions normally when set with the default value. When set to a value higher than 0, it only functions when kept pressed down for duration of the time set in seconds.

Commands activated in this mode (delay) exclude any normal commands eventually associated and vice versa.



This modality is not available in **webclient**; when a button is kept pressed down, all the delay commands that it has been associated with will be executed once only in sequence.

### 2.35.17. 3D Camera Viewer Commands

The "3D Camera" command settings for view activation are used for displaying previously stored 3D model scenes inserted on screen.

This command requires the 'views' of the 3D camera model in order to activate them and display the scene previously stored.

#### Call 3D Control

The name of the project's 3D object (or control) to be interacted with is entered here.

#### View Name

The name of the "View" to be activated is set here (needs to be recorded previously in runtime).

#### Delay Command Secs

This command functions normally when set with the default value. When set to a value higher than 0, it will only function when kept pressed down for duration of the time set in seconds.

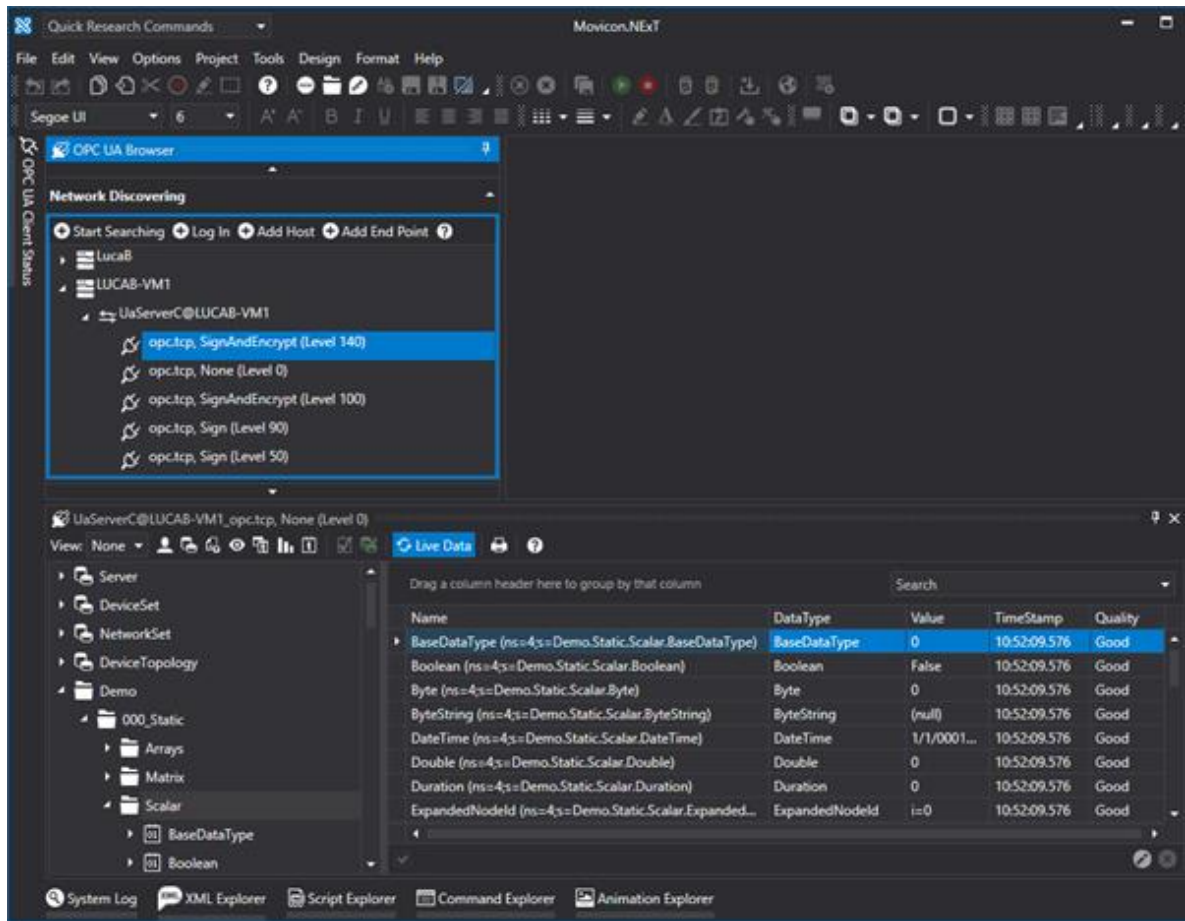
Commands activated in this mode (delay) exclude any normal commands that may have been associated and viceversa.



This modality is not available in **webclient**. When the button is kept pressed down, all the commands that it has been associated with will be executed once only in sequence.

## 2.36. Client Connectivity using an OPC UA Browser

The OPC UA Browser window allows you to connect and explore the resources exposed by the OPC UA Server in a network.



In the above screenshot shows an example of two Local Discovery Servers called "LUCAB" and "LUCAB-VM1". In the "LUCAB-VM1" Local Discovery Server a OPC UA Server has been recorded with the "UaServerC@LUCAB-VM1" Application Name which has five different Session Endpoints. Each Session Endpoint is identified by the triad: "Transport Protocol, Security Mode (Security Level)" for example "opc.tcp, SignAndEncrypt (Level 100)".

The OPC UA Browser presents a series of buttons to establish the connection with the OPC UA Server:

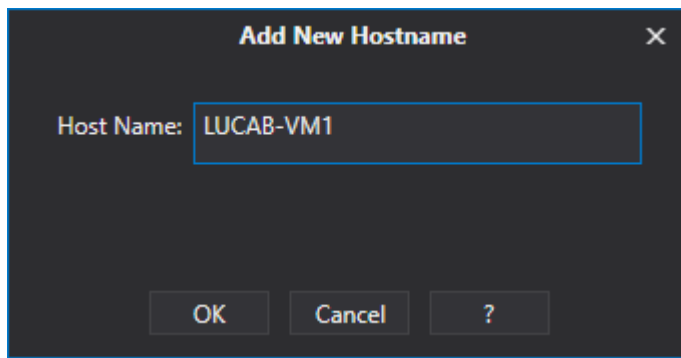
- Start Searching: scans the local network to check the presence of any OPC UA Local Discovery Servers.
- Login: permits User Login using their Credentials (username/Password) to the selected Endpoint, if not requested automatically in the connection phase.
- Add Host: allows you to define the Hostname manually (or IP address) of a Local Discovery Server from which to obtain all the Server's available Session Endpoints.
- Add Endpoint: allows you to manually define the Endpoint's url you wish to connect to.

### Endpoint connection

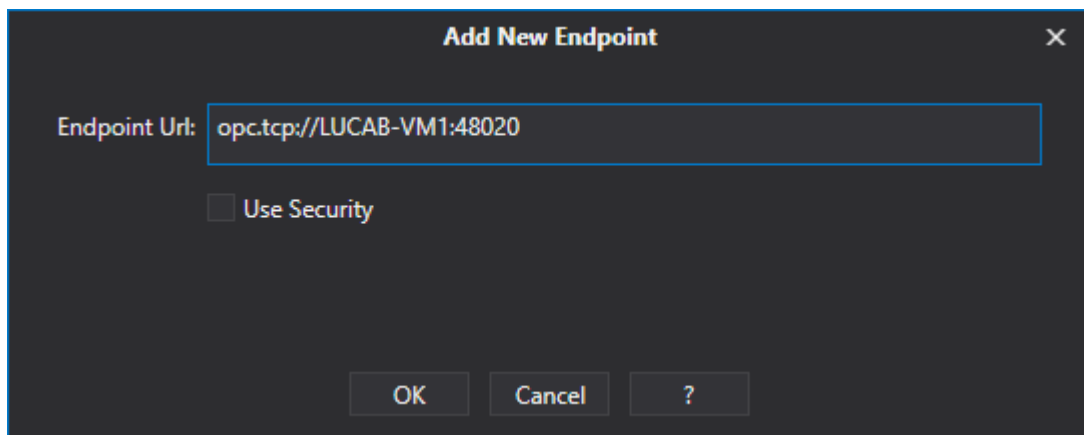
To end the procedure used to select an item from the OPC UA Server using the selection window, you will need to establish a Secure Channel by means of activating an Endpoint Session.

The Endpoint can be selected using one of three commands: "Start Searching", "Add Host" or "Add Endpoint".

A Host should be added if you do not know the Endpoints made available by the Server.



Otherwise, if you already know the Endpoint's url, you can select "Add Endpoint" to define the url using this format: "<Transport Protocol>://<hostname>:<port>", for example, "opc.tcp://server1:48020".



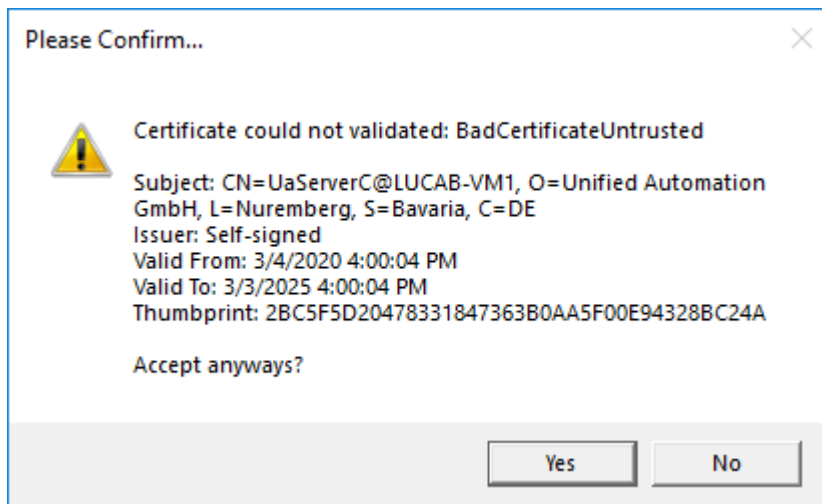
When you select the "Use Security" option, the Endpoint with a security level that is higher than those made available by the Server with a Sign or SignAndEncrypt Security Mode, will be selected.



**It would be preferable to use a hostname instead of an IP address to manually define the Host and Endpoint especially in those cases where the selected Endpoint uses the Sign or SingAndEncrypt Security Modes. Creating a Secure Channel, in these cases, involves the exchanging of public certificates between the Client and Server. The necessity to use a hostname (or IP address) depends on how the OPC UA Server's security certificate to be connected to has been defined.**

After having added an Endpoint or a Host that exposes several Endpoints, you will need to select one from the tree structure on the left with a double click to activate the communication channel towards the OPC UA Server.

If the security mode is Sign or SignAndEncrypt type, the Channel will open after the Application Instance Certificates have been exchanged. The Movicon.NExT OPC UA Browser receives and accepts the OPC UA Server certificate interactively by displaying a dialog window as shown in the screenshot below:



In this way the Movicon.NExT development environment, acting as Client, will accept the validity of the certificate for whole duration of the session.



**The OPC UA Browser can be made to accept Untrusted certificates automatically by setting this option:**

**<AutoAcceptUntrustedCertificates>true</AutoAcceptUntrustedCertificates>**  
in the file "OPCUAClient.Config" present in the Movicon.NExT installation folder

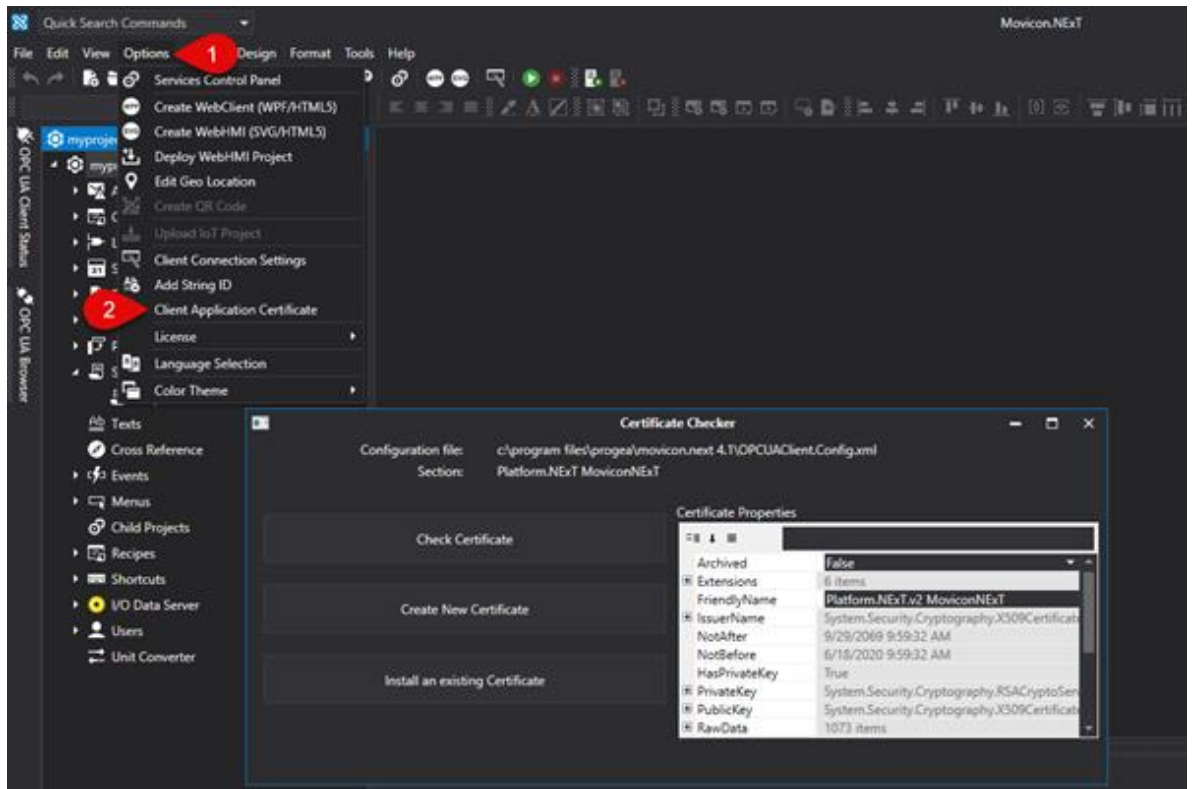


**In addition to receiving the certificate containing the Server's public key, Movicon.NExT also returns the certificate back to the same Server. However, according to the configuration, the OPC UA Server might refuse the certificate and place it in its list of rejected certificates. If this happens you will need to move the certificate from the list of Rejected Certificates and put it in the list of Trusted Certificates. Otherwise you can copy the Movicon.NExT certificate file to the Server according to the relative procedures.**

**The certificate to be used in this case is the "Platform.NExT.v2 MoviconNExT" file which is located in the "%ProgramData%\OPC Foundation\CertificateStores\MachineDefault\certs\" directory**

### Certificate management

The certificate of the Client part of Movicon.NExT called "Platform.NExT.v2 MoviconNExT" can be checked, renewed or replaced using the configuration tool accessible from the menu > option and selecting the Application Certificate item (2).



## 2.37. OPC UA Client Status Window

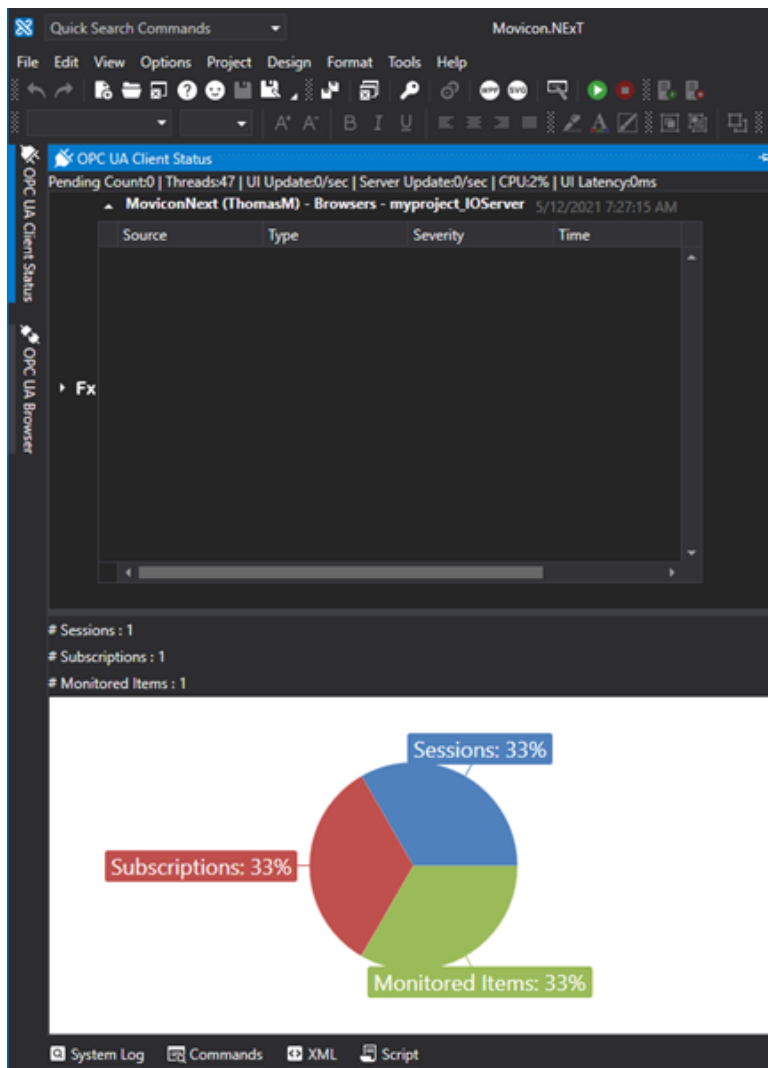
The "OPC UA Client Status" window reports information belonging to the OPC UA communications between the a Movicon.NExT OPC UA Server and Client, whether it be a client relating to the project or a client such as the OPC UA Browser window.

The OPC UA sessions open between the Client and Server and a series of information, which includes Threads and percentage of occupied CPU, are reported at the top of this window.

The following are reported at the bottom of the window:

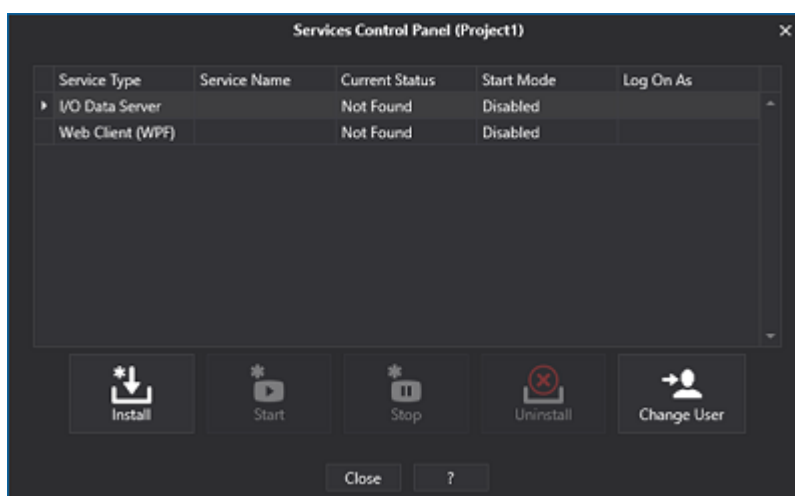
- The number of OPC UA communication sessions that the Client has opened towards the Server.
- The number of subscriptions that the Client has activated with the Server. The subscriptions are used to define data acquisition from the server.
- The number of Monitored Items, being the quantity of Tags the Client has subscribed to the Server.

The percentage of Monitored Items, Subscriptions and Sessions are shown in the pie chart compared to the grand total.



## 2.38. Services Control Panel

In the Options> service control panel menu there is a command, which opens a Window in which it is possible to manage the installation and uninstall of all the services of the project..





Only the services defined for the relative project sources are listed in the window.

In particular, the services relating to the Scripts and Logic are only available when the Script or Logic resources, which have been configured for Server startup in the project's properties, are active.

The Movicon processes that can be started up as Windows Service are:

- I/O Data Server: Project's IO Server
- Alarm Dispatcher: process to send alarm notifications by email, sms and other.
- Scheduler: process for executing time or event scheduled commands.
- Logic: process for executing the project's logic resource.
- Script: process for executing the project's Basic Script resources.
- Recipe: process for executing the project's Recipe resources.
- Web Client: process for executing the WebClient Server to connect using apps.

This window shows a series of status information (service name, status, etc.) for each service and commands for installing and running the various services.

Each service can be installed using a specific user. When a user is not specified, using the 'Change User' command before installing the service, the service will be installed with the machines "LocalSystem" default user.



In order to use another user that is not the "LocalSystem" default user, they must have access rights to 'Access as Service' in the operating system. This setting can be configured in the operating system by accessing the 'Control Panel - Administrative tools - Local Security Policy'. In the window that opens, select 'Local Policies - User Rights Assignment' and go to the 'Access as Service' item on the right and add the user desired to startup the process as service.



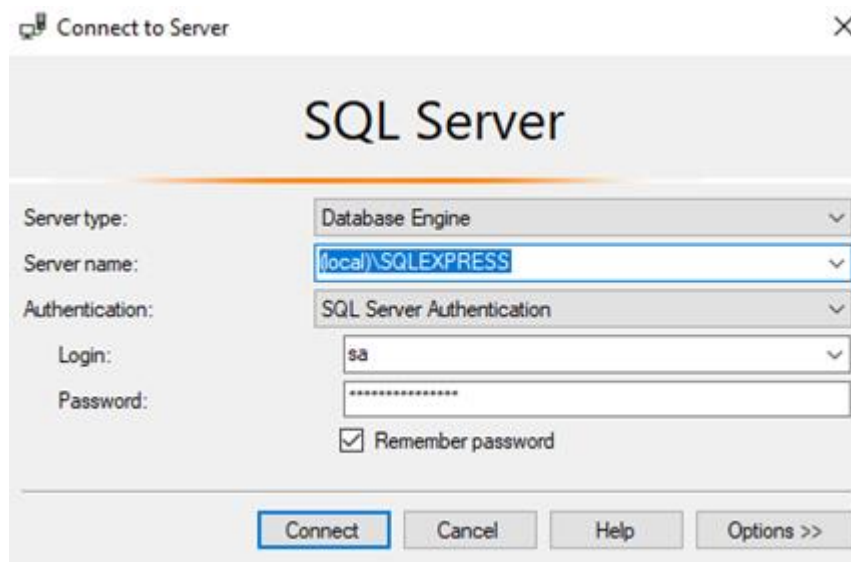
All the above listed services, except for the Recipe service and the Web Client (WPF) service, are dependent on the "I/O Data Server" service. This means that in order to startup these services, you will need to startup the "I/O Data Server" service as well.

### SQL Database Access with Movicon.NExT installation as Windows service

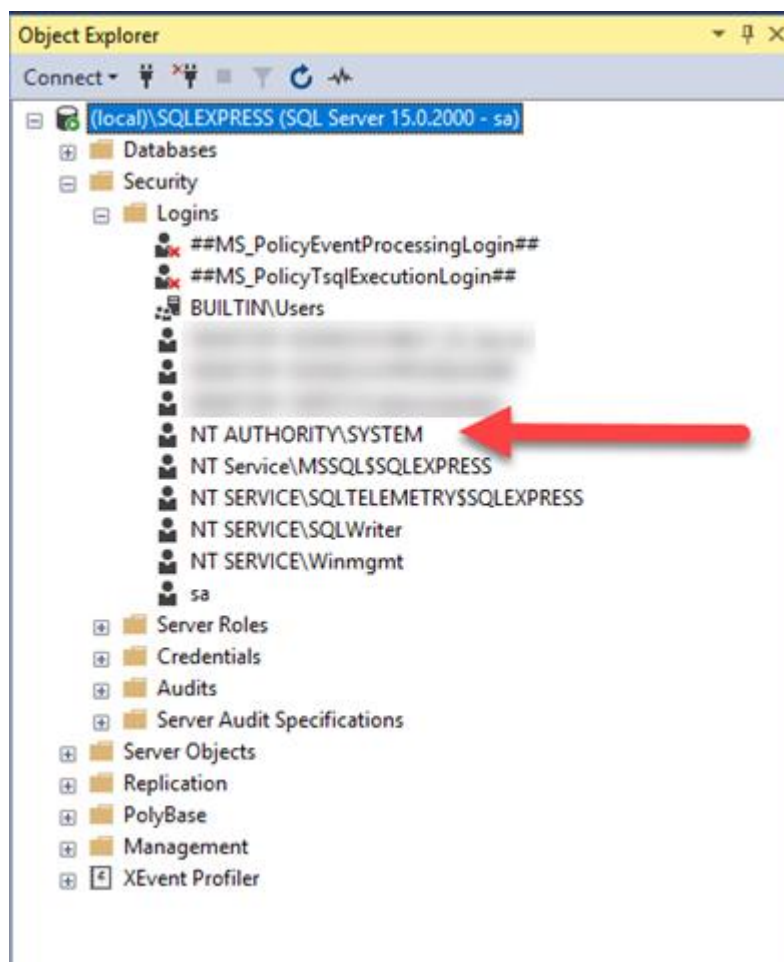
When wishing to install and run the Movicon.NExT process, such as the I/O Data Server and/or the Movicon.NExT WebClient WebHMI for example, so that they startup as Windows services, you must make sure that the Windows LocalSystem user, who is the one under which the services are installed and started up for default, has ownership of the Movicon.NExT progetto database project.

The simplest way to do this is to use an external tool such as Microsoft Management Studio. To activate the requested authorizations proceed as follows:

1. Open the Microsoft Management Studio and access with "SysAdmin" credentials:

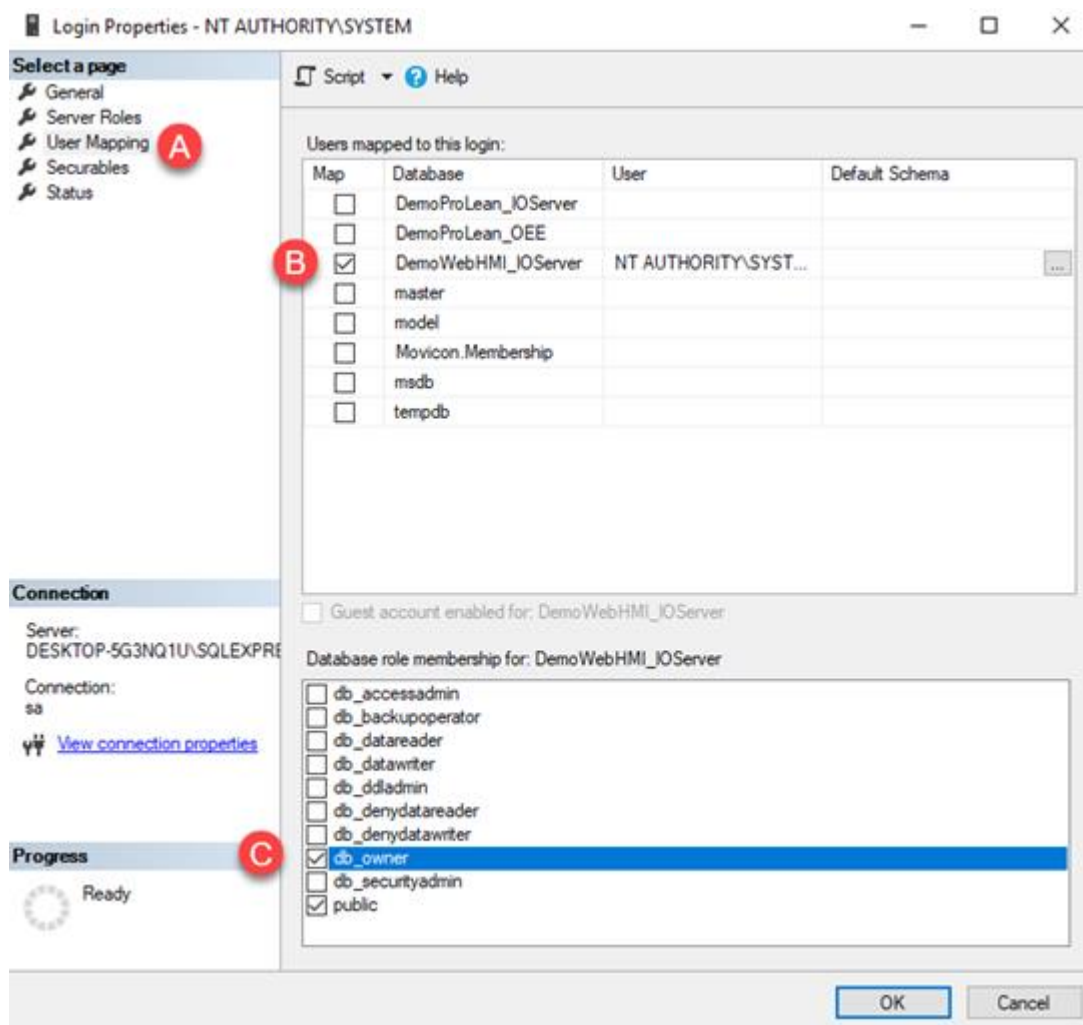


2. Open the Security\Logins folder in the Object Explorer pane on the left side.



3. Double click on NT AUTHORITY\SYSTEM user, the identity used in SQL Server for the Local System Windows user and open it:
  - select "User Mapping"
  - select the database in which you wish to map the user
  - assign the "db\_owner" membership role





Click OK to confirm.

## 2.39. Dynamic Configuration - Online Changes

The Movicon.NExT editor allows you to develop/edit a project during its Runtime without causing any impact on the application's operativity.

Below are listed the operating modes according to the nature of the changes and areas involved.

	Online changes supported by Clients and Servers on the same PC	Online changes supported by Clients and Servers pm different PCs
<b>Screen</b>	Yes	Yes
<b>I/O Data Server</b>	No, must restart Runtime	No, must restart the I/O Data Server
<b>Alarm Dispatcher</b>	No, must restart Runtime	No, must restart the Alarm Dispatcher
<b>Logic</b>	No, must restart Runtime	No, must restart Logic
<b>Schedulers</b>	No, must restart Runtime	No, must restart the Scheduler Service
<b>Report</b>	Yes	Yes

Parameters	Yes	Yes
<b>Script (Resource)</b>	No, must restart Runtime	No, must restart Scripts
<b>Script (in Tags)</b>	No, must restart Runtime	No, Necessario il riavvio dell' I/O Data Server
<b>Script (in screens and objects)</b>	Yes	Yes
<b>Text</b>	No, must restart Runtime	No, must restart Client
<b>Events</b>	No, must restart Runtime	No, must restart Client
<b>Menu</b>	No, must restart Runtime	No, must restart Client
<b>Recipes</b>	No, must restart Runtime	No, must restart the Recipe Service
<b>Keyboard Commands</b>	No, must restart Runtime	No, must restart Client
<b>Users</b>	No, must restart Runtime	No, must restart Client
<b>Unit Converters</b>	No, must restart Runtime	No, must restart Client

## Screen

Changes to Screens do not require Client restart and are applied when the screen concerned is opened. These changes, for example, include adding screens, adding/removing objects from the Toolbox or symbols in screens, changing animations or commands in Screen objects.

Exceptions:

- When the changes concern a currently displayed screen, simply update the screen by changing page and then returning back to it.
- When the changes concern a Screen with its ""Keep in Memory" parameter active , or has been preloaded at startup, you will need to restart the Client.

## I/O Data Server

Changes to the project's I/O Data Server require you to restart the I/O Data Server.

Example: adding Drivers, adding Tags, creating/editing Alarms, creating/editing Dataloggers or Hlstorians and so forth.

When the project has been started up with the "-start" option or from the development environment, you will need to restart the entire project. On the other hand, when the component has been started up as service, you will only need to restart the service concerned while keeping the rest of the project running. The Redundancy option can be aquired to avoid causing any impact on the application's operativity when starting up the I/O Data Server.

## Alarm Dispatcher

Changes to the Alarm Dispatcher require Alarm Dispatcher restart.

Example: adding new plug-ins, adding/removing/editing notifications and other. When this component has been installed and started as server, you will only need to restart the service concerned while keeping the rest of the project running. On the other hand, when the project has been started up with the "-start" option, or from the development environment, you will need to restart the entire project.

## Logic

Changes to logic require that the logic be retarted.

Example: adding a new Logic resource or changing a logic resource and so on.

When this component has been installed and started up as service, it will be possible to restart on the service concerned while keeping the rest of the project in runtime.

On the other hand, when the project has been started up with the "-start" option, or from the development environment, it will be necessary to restart the entire project.

## Schedulers

Changes to Schedulers require their restart.

Example: adding new schedulers, changing schedulers and so on.

When this component has been installed and started up as service, it will be possible to restart just the service concerned while keeping the rest of the project in runtime.

When the project has been started up with the "-start" option, or from the development environment, it will be necessary to restart the entire project.

## Reports

Changes Reports do not require you to restart Client and are applied upon opening the reports concerned.

Example: adding Reports, changing the content of Reports, changing the Report queries or parameters and so on.

When the changes concern the currently displayed Report, you will only need to close and reopen the Report.

## Parameters

Changes to Parameters do not require Client restart.

Example: adding parameters, editing contents of a parameter and so on.

When changes concern a parameter used by a currently displayed screen, you will need to update the Screen by simply changing pages and returning back to it.

## Script (Resource)

Changes to Resource Scripts require Client restart.

Example: adding Resource Scripts, editing Resource Scripts and so on.

When this component has been installed and started up as service, it will be possible to restart just the service concerned while keeping the rest of the project in runtime.

When the project has been started up with the "-start" option, or from the development environment, it will be necessary to restart the entire project.

## Script (in Screens/Objects)

Changes to Scripts contained in Screens or Objects do not require Client restart and are applied when the concerned Screens are opened. Exceptions:

- When the Script is kept active in a Loop.
- When changes concern a Script in a currently displayed Screen, you will only need to change page then return back to screen to update it.
- When changes concern a Script in a Screen that has an active "Keep in Memory" parameter or that has been preloaded at startup, it will be necessary to restart the Client.

## Script (in Tags)

Changes to Scripts in I/O Data Server Tags require you to restart the I/O Data Server.

When the project has been started up with the "-start" option, or from the development environment, it will be necessary to restart the entire project.

When this component has been installed and started up as service, it will be possible to restart just the service concerned while keeping the rest of the project in runtime.

In addition, the Redundancy option can be acquired to avoid causing any impact on the application's operativity when starting up the I/O Data Server.

## Text

Changes to the Text table require Client restart.

Example: adding a new language, adding a new String ID, changing String ID translations etc...

When the project has been started up with the "-start" option or from the development environment, you will need to restart the entire project.

When the Client has been started up with the "-start -client" option, you will only need to restart the Client while keeping the rest running.

## Events

Changes to the Events table require Client startup.

Example: Adding an Event, changing an event, etc...

When the project has been started up with the "-start" option or from the development environment, you will need to restart the entire project.

When the Client has been started up with the "-start -client" option, you will only need to restart the Client while keeping the rest running.

## Menu

Changes to Menus require Client startup.

Example: adding a Menu, changing a Menu, etc...

When the project has been started up with the "-start" option or from the development environment, you will need to restart the entire project.

When the Client has been started up with the "-start -client" option, you will only need to restart the Client while keeping the rest running.

## Recipes

changes to Recipes require Recipe restart.

Example: adding a Recipe, changing a Recipe, etc...

When this component has been installed and started up as service, it will be possible to restart just the service concerned while keeping the rest of the project in runtime.

When the project has been started up with the "-start" option or from the development environment, you will need to restart the entire project.

## Keyboard Commands

Changes to Keyboard Commands require Client restart.

Example: adding a new Keyboard Command, changing a Keyboard Command, etc...

When the project has been started up with the "-start" option or from the development environment, you will need to restart the entire project.

When the Client has been started up with the "-start -client" option, you will only need to restart the Client while keeping the rest running.

## Users

Changes cannot be made to Users and User Groups in the development environment while the project is running.

However, users can be managed at Runtime using the "Edit User" command.

## Unit Converters

Changes to Unit Converters require Client restart.

Example: adding a Unit Converter, changing a Unit Converter, etc...

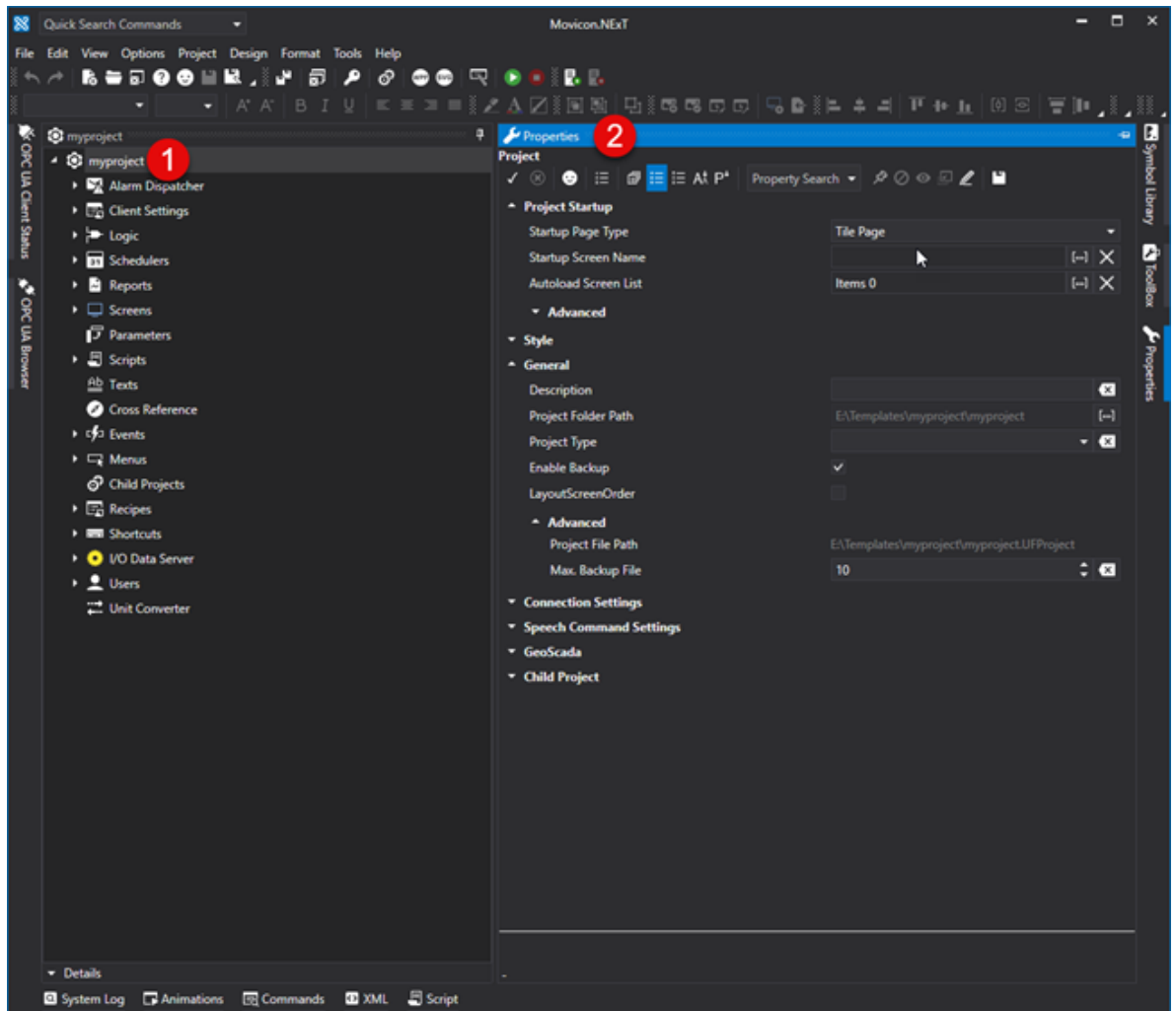
When the project has been started up with the "-start" option or from the development environment, you will need to restart the entire project.

When the Client has been started up with the "-start -client" option, you will only need to restart the Client while keeping the rest running.

## 3. Editing Projects

### 3.1. Project Properties

The properties described below are not just specific to one individual resource but are common to all the project resources. Part of these properties, such as the Geo-Localization, are also displayed in other resources such as screens.



#### General

Below are listed the general project properties.

#### Project File

Indicates the complete project path (not editable).

#### Project Folder Path

Indicates the project folder path (not editable) .

#### Autoload Screen List

Used to set the list of screens to be automatically loaded at project startup.

### Description

This is used to enter a brief project description.

### Enable Project Backup

When this property is enabled, a backup copy will automatically be created when using the "Save All" command.

When the project is in "xaml" file format, it is also possible to select the number of backup files (in ".zip" format) that should be maintained. Once this set number has been reached, the creation of a new backup file will eliminate the oldest one.

When the project has been saved on DataBase, it will be possible to create only one backup file which will be overwritten each time a backup is made. The file is found in the SQL Server Backup folder, in the folder of the version used. For example of this path might be:

`C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Backup`

### Reload Last Screen Opened and and Zoom

When setting this property to true, the last screen opened by the user will be reloaded resetting the zoom state and position of the Scrollbar (usable only on WebHMI side).

### Max. Backup File

Sets the maximum number of backup files to be saved. Once this number has been reached, the project will start overwriting the oldest ones.

### Culture Name

The default Culture will activate at the first project startup. Subsequent project startups will restore the last language activated before the project closed.

### Converter Name

The default Unit Converter which will activate at the first project startup. Subsequent project startups will restore the last Unit Converter activated before the project closed.

## Startup Project

Below are described the properties which are used during the project startup.

### Startup Page Type

Select the type of page to use at the project startup (Main Screen, Tile Page, Geo Page).

### Autoload Screen List

This is used to set the list screen that will automatically load at project startup.



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

### Startup Logics List

This is used to set a list of logic to be executed automatically at project startup.

### Startup Scripts List

This is used to set a list of scripts to be executed automatically at project startup.

### Startup Screen Name

This is used to set which screen is to open at project startup and requires that the 'Startup Page Type' be set to 'Main Screen'.

### Connect Items at Startup

When enabling this property at Client startup, the Server Tags will be connected gradually including those not in use in the Client (with default time set in the settings). This will quicken the subsequent opening of screens or script executions that put variables in use for the first time.

## Style

The Style properties are those which determine the project's general aesthetic characteristics.

### Project Theme

This property is used to set the project's internal theme graphics for designing the various window and toolbar styles.

### Touch Control Type

This is used to set the touch technology type (if supported) to be used in the project.

### Background Color

This property is used to set the background color of the "Main Page". This property is only used if the "Startup Page Type" has been set with "Main Page".

## Child Project

This property group contains the general settings for Child projects.

### Child Visible

Enabling this property will show the child project screen in the "Main Page".

### Child Tile Size

Sets the size of the tile where the child project screens will be displayed in the "Main Page".

## GeoScada

This group of properties concern the Geo-Localization feature.

### GeoMap Type

This property is used to set the geographical map type to be used. This property can only be used when the "Startup Page Type" has been set with the "Geo Page" option.

## Voice

This property group contains some general settings for the voice controls.

### Enable Speech Command

This property is used to enable the Speech Command function.

### Speech Confidence

Increases or decreases Speech Command recognition sensibility of the voice Controller.

## Connection Settings

The connection properties determine some of the modalities relevant to project data connections and are grouped in the following settings:

### Update Time

This value added to the "Item Removal Delay" property indicates after how many seconds to download project and place any variables present within project not in use.

### Disable When Not In Use

Enabling this property, the Tags that are not in use by the Client are forced to switch to the "Non Active Item" state and therefore will be removed when time set in the "Item Removal Delay" property expires.

### Non Active Item Update Time

This is used to set an update time frequency for existing OPC ITEMS that are going into 'not in use' state. This parameter is passed to the Server and any Driver when downloading project.

### Active Item Update Time

This is used to set an update time frequency for existing OPC ITEMS in use. This parameter is passed to the Server and any Driver when loading project and variables go into use.

### Use Secure Connection

This permits only to connect to secure server connections declared safe by the appropriate certificates.

### Number of Items to Remove

This property is used to specify the number of OPC items to set to the "Inactive Item" state and removed as a consequence each time the time defined in the "Item Removal Delay" property expired. This value is only considered when the "Disable When Not In Use" property is disabled.

### Item Removal Delay

This property is used to specify a delay time with which OPC UA subscriptions of items in the "Inactive Item" state will be removed. Value is expressed in seconds.

## 3.2. Project Folders and Files

The various information on Movicon.NExT installation, folder and file usages are saved in the following paths:

### Movicon Installation Folder:

%ProgramFiles%\Progea\Movicon.NExT x.x\

### Folder of Symbols from the Toolbox:

%ProgramData%\Progea\Movicon.NExT.x.x\Toolbox\

### Folder of Symbols from the Library (System) Symbols Folder:

%ProgramData%\Progea\Movicon.NExT\Symbols4\



#### Folder of Symbols from the Symbol Library (Project):

...\<ProjectName>\<ProjectName>\Symbols\

#### Editor's Storage Settings Folder:

%LocalAppData%\IsolatedStorage\

#### License File storage Folder:

%ProgramData%\Progea\Movicon.NExT\

#### Help file storage Folder:

%ProgramData%\Progea\Movicon.NExT.x.x\HelpOnLine\

#### Symbol localization file storage Folder:

%ProgramData%\Progea\Movicon.NExT.x.x\Cultures\

#### Screen Template storage Folder:

%ProgramData%\Progea\Movicon.NExT.x.x\NewScreenTypes\



Conventions:

%ProgramFiles% = usually C:\Program Files

%ProgramData% = usually C:\ProgramData

%LocalAppData% = usually C:\Users\<User\_Name>\AppData\Local

<ProjectName> = Name assigned to Movicon.NExT project

## 3.3. Text and Multilanguages

### Introduction to using Text

Generally a graphical interface project has to display and manage text strings to be displayed in screens as simple text and as headings, names and text in objects.

The text strings can be freely typed in any object directly by clicking on the area provided, or entered in the property window fields. Texts in screens can also be managed using the **"Text"** viewer object available from the basic shapes section in the **Toolbox**.

Some Movicon.NExT objects support the association of texts to be displayed as object titles (e.g. the 'Text' object, or Buttons). In this case static text or String ID can always be associated. The use of String ID is the most preferred because not only does it guarantee the possibility to change languages during runtime, it also allows text to be kept centralized in one container known as the String Table.



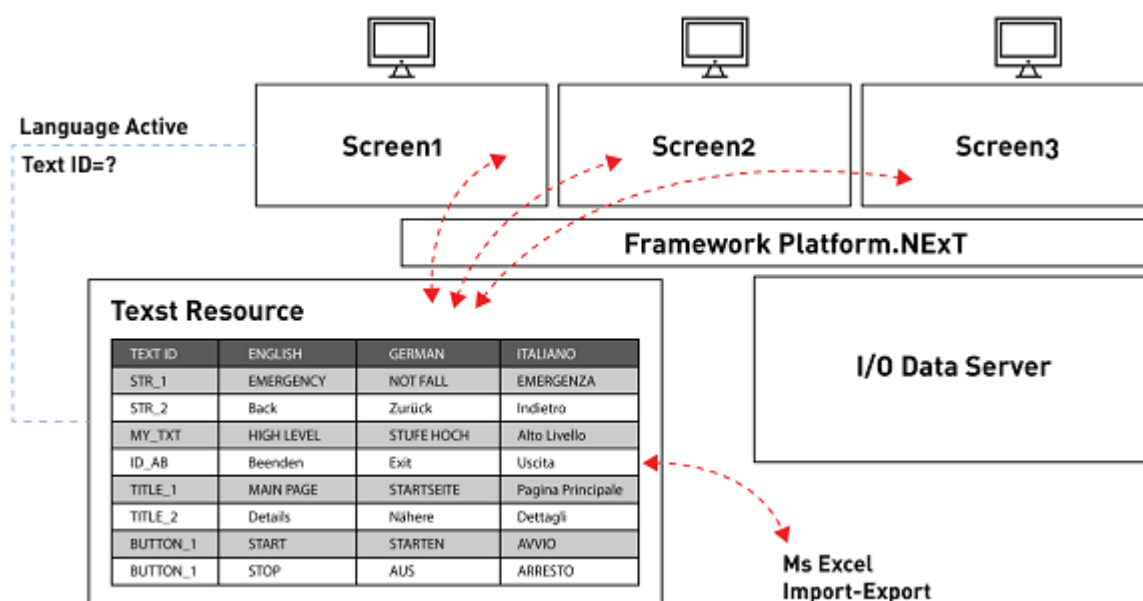
For further information about static text in objects please see the topic on Objects from the ToolBox.



In all circumstances, text in Movicon projects is managed either in static or dynamic mode, according to whether the text is inserted as an **ID identifier** in the text table of the project's String Resource.

## The String Resource

The Resource of the Project's String Tables (Texts) is the container of all the texts that can be dynamically managed and organized in the Movicon project. The String Table is a text table organized in columns which are compatible with Excel and correspond to the ID identifier texts and their languages as shown in the image below:



For each text string inserted on screen, Movicon searches for its corresponding text as an "ID identifier" in the String Table to replace the ID Text with the text contained in the corresponding language column. In this way, by using a univocal string id Movicon will be able to manage the dynamic language change function in runtime by displaying the text contained in the column of the active language.



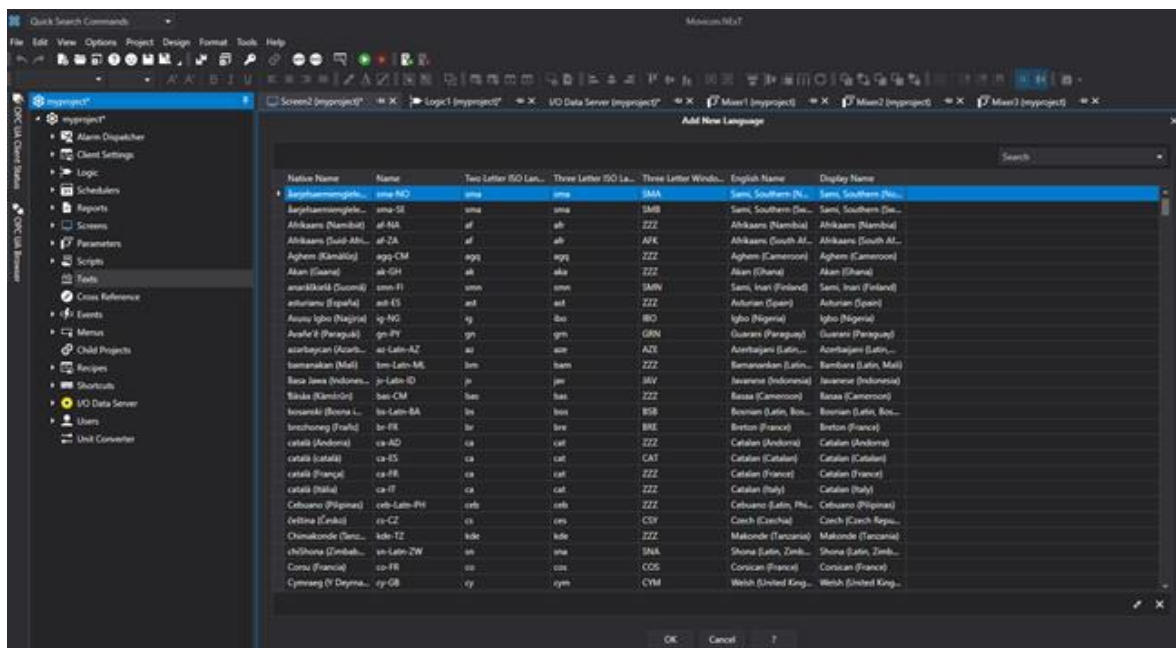
A "Change Language" command can be assigned to any command object to allow the user to active the column of the language desired. All the Movicon texts will dynamically update as a consequence.

The text Strings are needed dynamically managing for example alarm text, headings of objects or windows, any text used in screens or controls that provide use of text. When there is the possibility to manage text in the project's user interface, this text can be kept in the String Table and therefore centralized in one point to be managed dynamically in function with the language desired.



Remember that any text entered directly in Movicon drawings or components without any corresponding ID existing in the String Table will be represented how it was written in static mode in the source representing them and therefore will not be subject to change language.

To open a project String editor double click on the "Texts" in the Project Explorer Window.



*The String Editor window appears in table format with an "ID" column and a column for each language inserted*



Movicon provides the possibility to load the string table from custom files inserted in the project's string folder. For further information please see the topic on: Runtime Strings.



It is possible to use the following "\n" escape sequence to force the text to go to the beginning of the next line. For example the string:

Item 1\nItem 2\nItem 3

in the visualization phase, for example a text box or button will appear as follows:

- Item 1
- Item 2
- Item 3

## Text Management

When opening the String Resource for the first time you will need to create at least one column relating to a text language using a window which automatically opens for selecting a language (or Culture). In this window you will find listed all the languages identified by the Windows operating system. Each one of these languages is represented with an identification that reports the language and geographical area. For example, "en-GB" and "en-US". In this case the first part of the identifier represents the

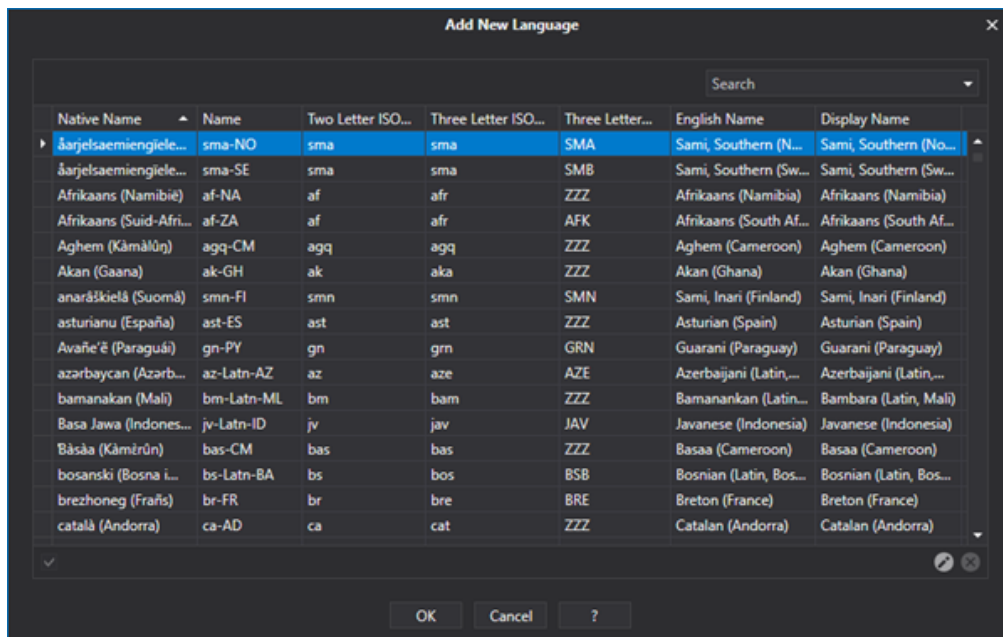
language (en = English) and the second part represents the geographical area (GB = Great Britain and US = United States).



The definition of the language name is preset by the operating system and cannot be changed. The language name is conventional and necessary to eventual automatic text translations as we will see below.

## Adding a new language

The use of the String Table requires that at least one text column be added referring to the selection of one language. The first time the resource is opened, the window for adding a new language opens automatically. Other new languages can be subsequently added by using the "Add New Language" command available from the Text Toolbar or with the right mouse key.



*The Add New Language command is used for adding new languages to the String Table*

This command can be used each time a new language has to be added to the project.



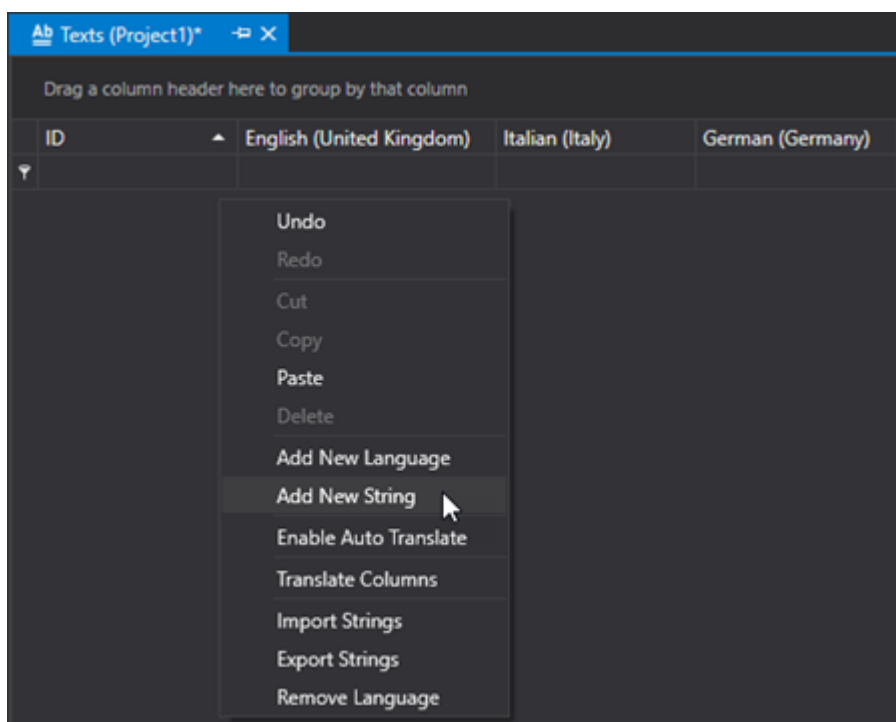
Movicon does not impose any limits on the number of languages that can be inserted in the project. However, the product may not be guaranteed to work correctly with projects using more than 32 language columns at the same time.

## Adding a new text string

After creating the Columns corresponding to the languages, you can add new text strings by editing the Text Table manually or by importing strings already used in the project into the Text Table.

### Manual Editing

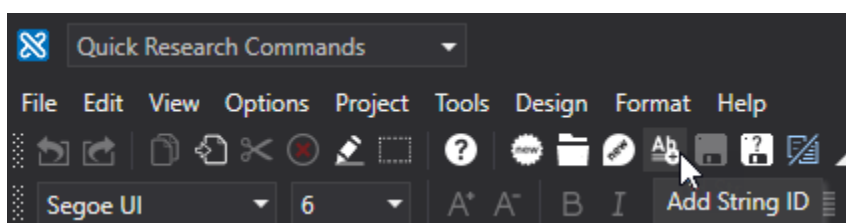
New text strings can be added manually by selecting "Add New String" from the contextual menu. To edit the text simply double click on the cell. To go from one cell to another on the table, you can either use the direction arrow keys or the tab key. Below is a screen shot of the string resource editor window.



### Importing strings used in the project

To add strings that have already been used in the project to the Text Table in order to translate them into different languages, use the Options menu > Add String ID, or the icon in the Main Toolbar.

According to which resource is open (Screen, Alarms...), the used strings will be imported to the Text Table.

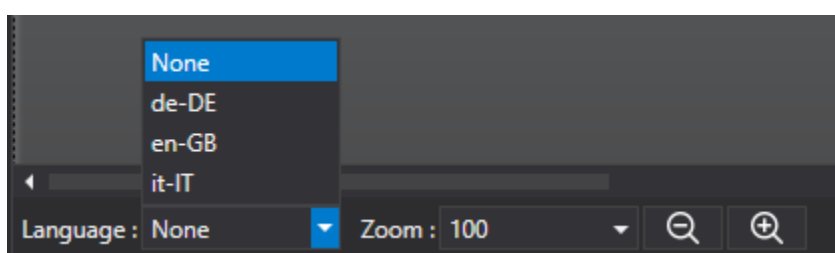


In addition, it is also possible to use the Cross Reference to analyze strings used in the project but not yet added to the Text Table. In order to do this, please refer to the topic on "Cross Reference" for further details.

## Managing Strings in the editor

Language settings can be managed within the development environment by using a popup menu placed at the bottom of the screen editing window.

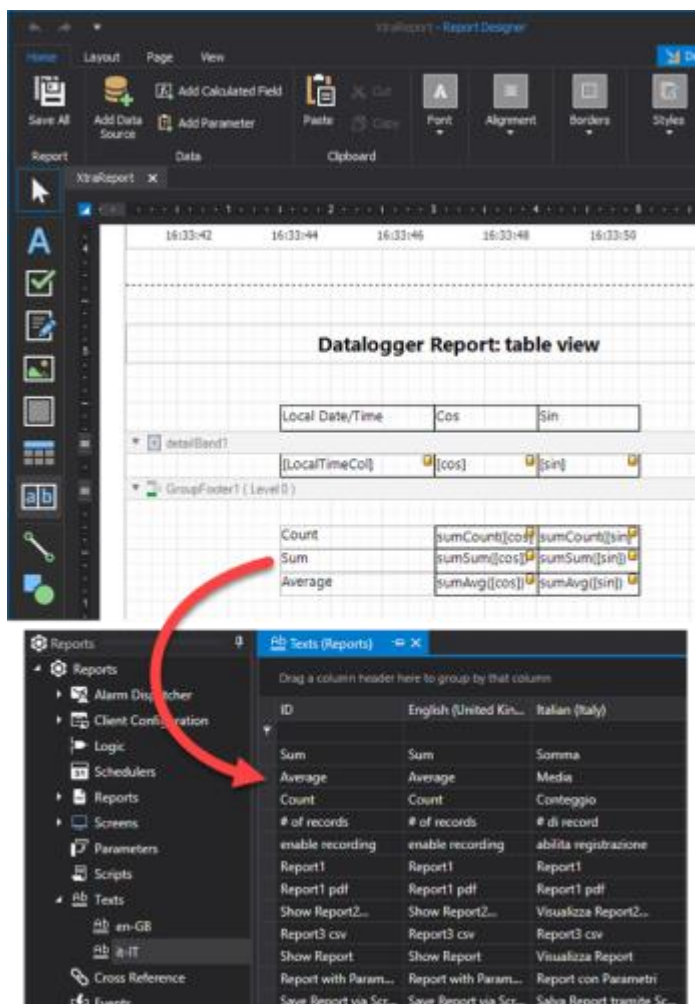
The languages to choose from are those set in the project.



When the project is started up in runtime without a selected language, it will start with the language selected in editing mode. Subsequently in runtime, any language set by using the command for objects will have priority.

## Strings Support in Reports

The text defined in the Label objects within Reports behave like "String IDs" within screens. Therefore, when "String IDs" that are present in the text table are used in a report's labels, they will be replaced with the language currently in use at Runtime, just like they do in the project screens.



## Editing Strings at Runtime

Movicon.NEXt provides the possibility to load the string table not from the table defined in design mode but from a custom file, if one exists, added to the project's strings folder. This will allow you to customize the project's strings without having to open them in design mode.

Both the Server and Client will search for the files in the project's "Strings" folder that have the "\*.custstrings" extension. When these files are found, they will be imported and used instead of the String Table defined in design mode.



In cases where there is at least one 'runtime' file, the table defined in design mode will not be loaded and as a consequence if those strings defined in design mode are missing in the runtime file, they will be displayed with the simple "String IDs" associated in the object's or resource's text.

The runtime file must have the same import file structure as for the string editor, being:

1° riga: ID;[cultura1];[cultura2],...  
next rows: Key;string1;string2,...

A practical file example would be:

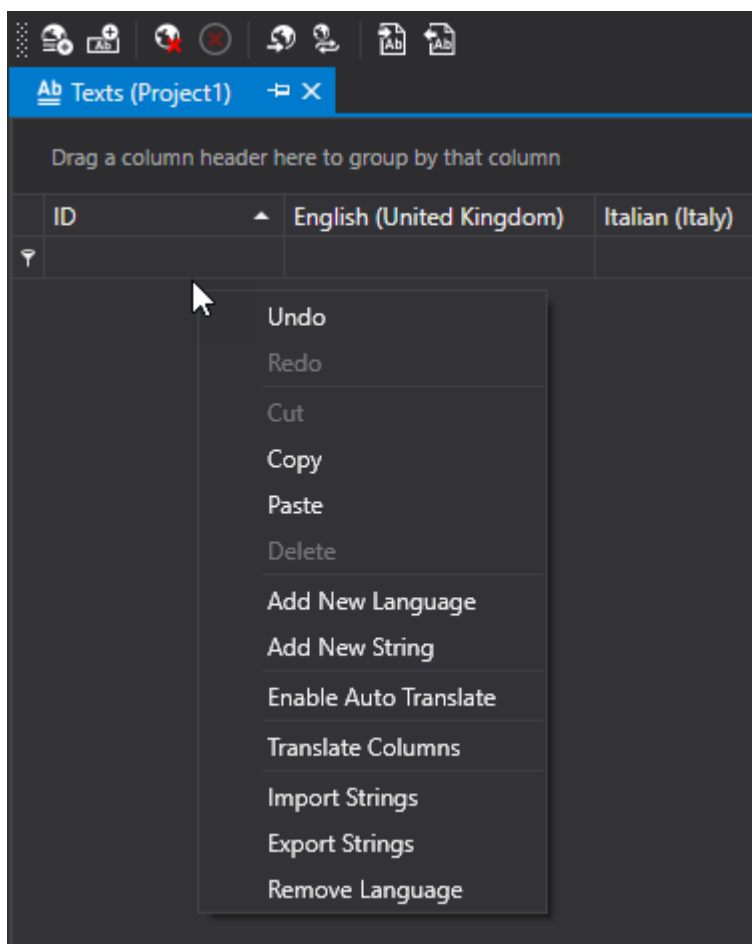
```
ID;it-IT;en-US;es-ES
NewTextID_01;Ciao;Hello;Hola
NewTextID_02;Allarme;Alarm;Alarma
```



The runtime strings are also loaded when the string table in design mode is empty. However, it is necessary that the table in design mode be defined with cultures you wish to use and that these cultures are also present in the runtime file.

## String Editor Commands

The command used for editing strings are available in the "Texts" toolbar and in the contextual menu.



### Add New Language

This command is used to insert a new Language Column in the string table. The available languages will be shown in a list (which can be put into order) contained in a pop-up window. Simply click on the language desired and confirm with the 'OK' button to add it to the table.

### Add New String

This command is used to set a new String ID. The string ID will be initialized with a default value that can be changed using the following modes:

- select cell desired and press the "F2" key on keypad
- double click on desired cell
- select cell desired and start digiting text using the keypad.

This command is equivalent to pressing the "Ins" key on the keypad.

### Enable Auto Translate

Movicon.NExT supports the automatic text translation function. This function can be used when wishing to have an automatic translation of text into any language inserted in the project.

This Movicon feature uses the automatic online translation services and therefore requires that the PC, being used for this operation, be connected to the internet.



The quality of the translation cannot be guaranteed depending on the automatic online translation services provided. Therefore it is advised to check translations with a translator when in doubt to the correct translation.

It will be the responsibility of the designer or translator, therefore, to evaluate the quality and correctness of the translation obtained, and modify it where necessary.

Translation operations are therefore manageable through the following concepts:

#### Active Automatic Translation

Movicon will automatically translate any text inserted in any column and insert translated text into empty columns.

#### Non Active Automatic Translation

No automatic translations will be performed

#### Translation Command

This command forces the translation of text in a column and requires that the language of origin and the destination language be specified (even when the Automatic Translation is not active).

- **ATTENTION: The translation command overwrites any previously modified text.**

### Translate Columns

This command is used to automatically translate all the strings declared in the table by using a pop-up window to select the language to translate and then the columns to translate this language.



Usually this command is used when adding new language columns to the already populated string table.



This feature requires an active Internet connection.

### Delete

This command allows you to delete the selected string from the string table. This feature supports multi-selection with "CTRL+click" and dragging techniques by selecting column row on the left hand side and dragging mouse cursor along the column. This command is equivalent to pressing the keypad 'Canc' key



## Remove Language

This command is used to remove language columns from the string table. This is done by selecting the desired column from the pop-up window that appears and confirming with OK. This command does not support multi-selections.

## Import Strings

This command is used to import text strings in CSV. file format. Movicon will import all the ID and language columns contained in the CSV. file to the string table.



The strings that already exist in the table will automatically be overwritten with those imported from the CSV file. Those language columns that have not been declared in the project but exist in the CSV file will automatically be inserted in the string table.

## Export Strings

This command consents to exporting text strings by saving them on file in CSV. format. This command supports single and multiple selections using "CTRL+click" and dragging the mouse cursor along the column containing the pointer to the selected row to the left of the "ID" column. It will be then possible to select and export all the strings from the table or only those selected in the export window.

## Special String ID

Special string ID can be inserted in the Movicon.NExT string table to customize the texts of various objects from the ToolBox in runtime. These particular string id comes in two parts. The first part contains the Place folder with the name of the control ( `_ {ControlName}_` ) and the second part contains the control's reference text to be customized e.g:

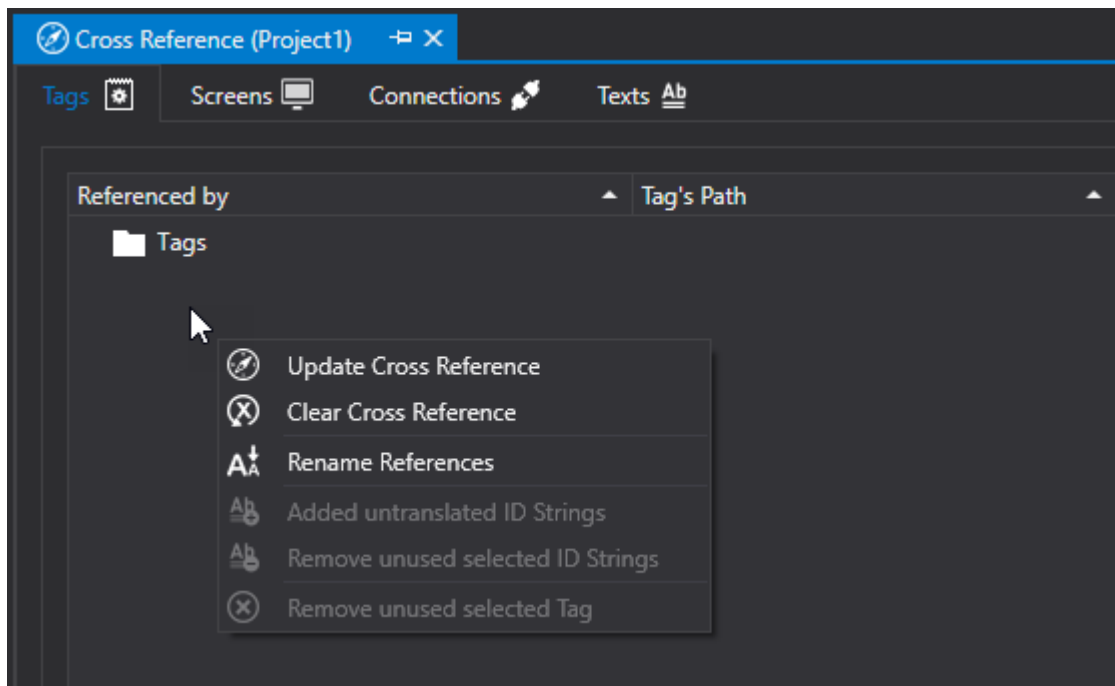
### \_AlarmWindow\_Message

In this case, the text to be inserted in the box that corresponds to the language will change the original text in the object when in runtime. In order to change the language in the project during runtime, you will need to use a change language command associated to a button.

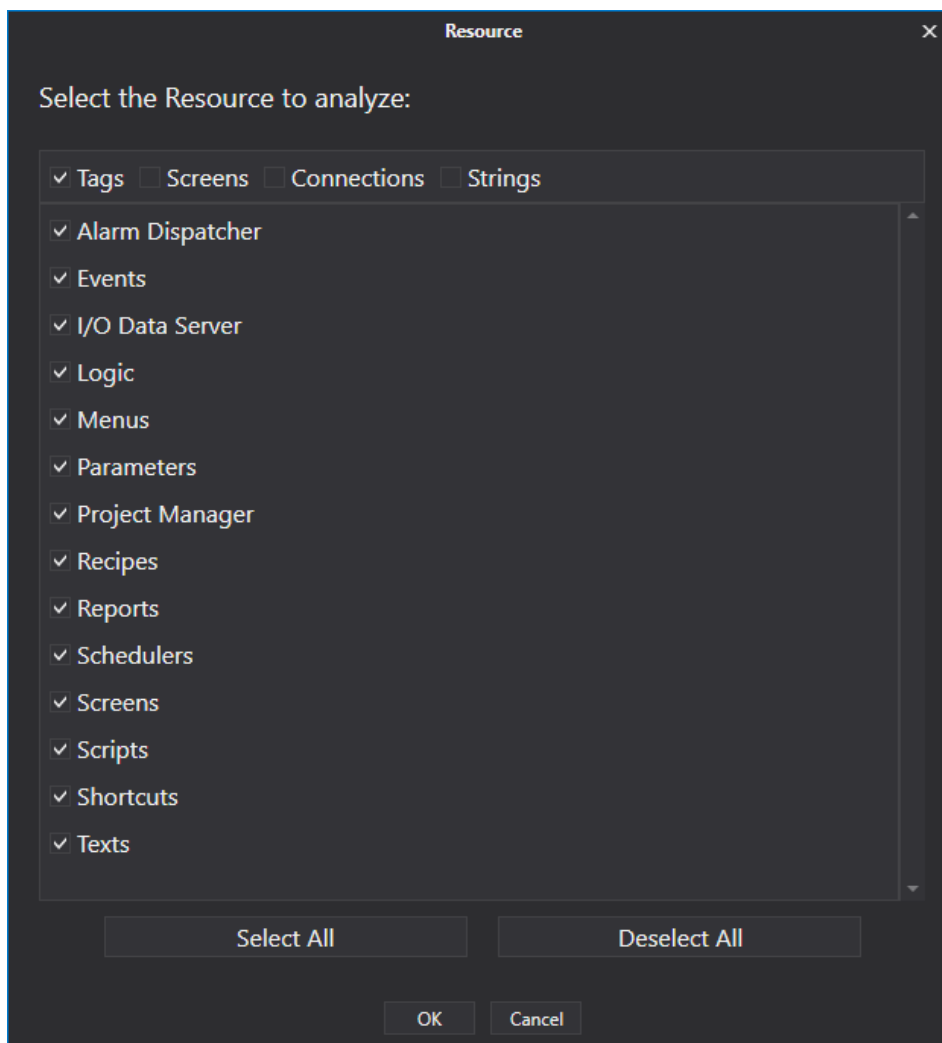
Ab Texts (Project1)		
Drag a column header here to group by that column		
ID	English (United Kingdom)	Italian (Italy)
_AlarmWindow_Message	Alarm Text	Testo Allarme

*The replaced "ITA mess4g3" text will show in the AlarmWindow object's message column when changing the project language to Italian (Italy).*



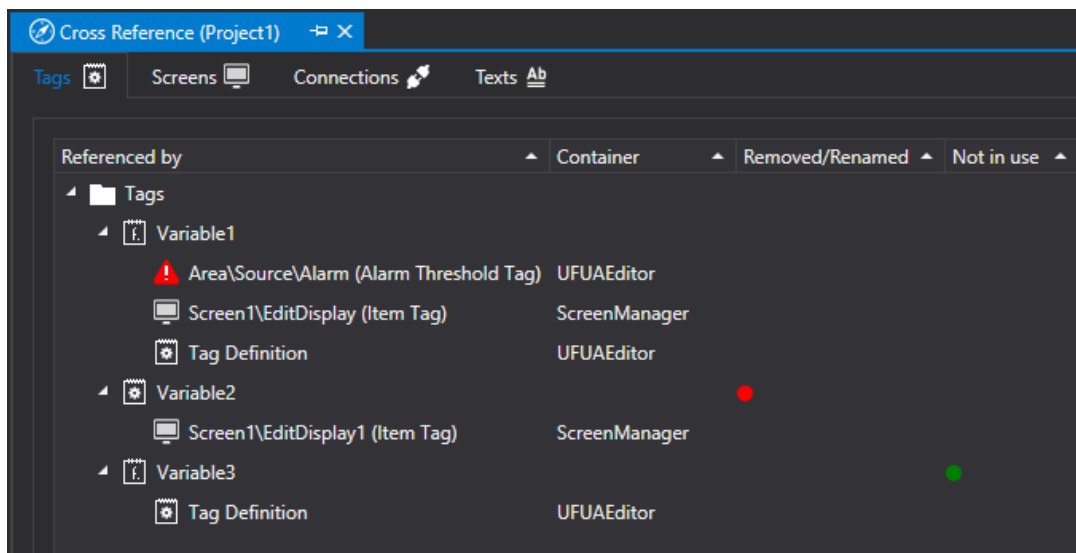


a window will open with the possibility to select the resources for which you wish to compile the cross reference.



- Keep in mind that the command performs an analysis on the project's current situation. Therefore any modifications made afterwards will not be updated in the Cross Reference until the "Update Cross Reference" command is used again.
- Variables which are detected in scripts will only be those used with direct notation. Other variable types, such as those retrieved with the `GetVariable` and `SetVariable` functions, will not be detected.

The window reports the information for the resources of interest as shown by the screen shot below:



According to the Resource selected, such as: Tag, Screen, Connection and Text, it will be possible to display the map of occurrences and double-click on each one to access the object which it references.

The "Removed/Renamed" column indicates with a red icon the occurrences of the resource that have been Renamed or Removed but still referenced with its old name.

By means of the "Rename References" contextual menu or the Project > Cross Reference > Rename References menu, it is possible to align the name of each resource with those in the one used in the different object connected to it.

If the red icon remains after executing the "Rename References" command, this will mean that the resource (Tag, Screen...) has been cancelled but is still being used in some of the project's objects.

The "Not in Use" column indicates with a green icon that the resource of interest is not currently in use in the project.

When the resource is a Tag, you can use the "Remove Tags not in use" command to eliminate the resource from the I/O Data Server.

Other commands available in the contextual menu are:

### Clear Cross Reference

This command is available on the Cross Reference Toolbar or the contextual menu, or the Project menu > Cross Reference > Clear Cross Reference.

### Add Untranslated Strings

This is used to add the selected texts that have not been translated to the Texts (Strings and Multilanguage) resource.



This command is available only for the Texts Resource.

### Remove Strings not in use

This is used to remove the selected texts that are not in use in the project from the Texts (Strings and Multilanguages) resource.



This command is available only for the Texts Resource.

## Cross Reference at Runtime

You will find a 'Cross Reference Viewer' in the Toolbox that will allow you to access the cross reference from the project at runtime.

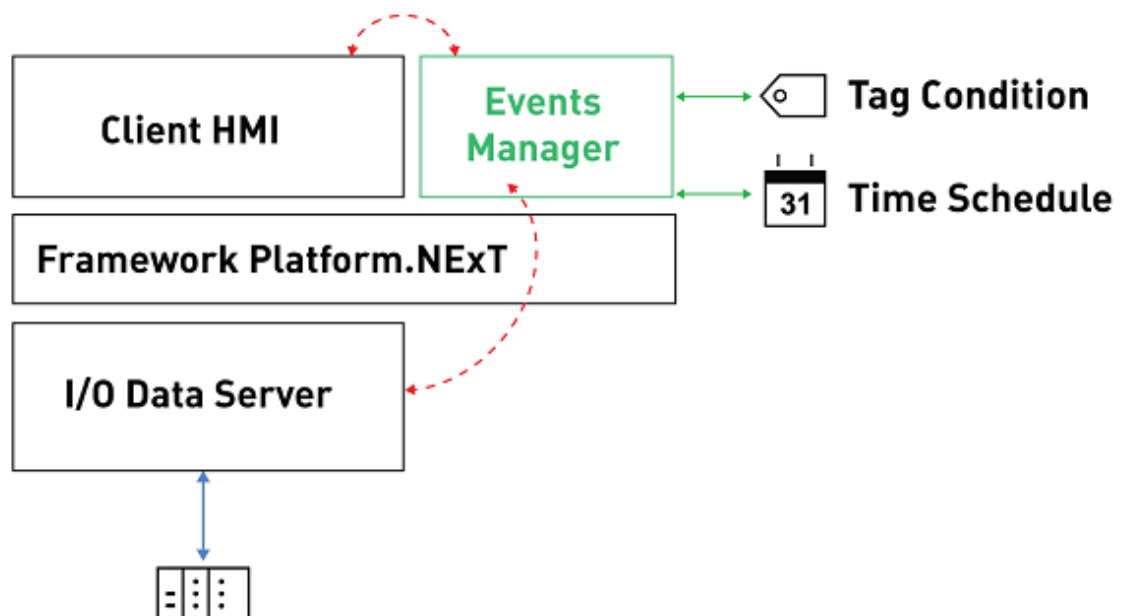
In addition, for debug purposes, you will be able to access the 'Cross Reference' using the "Shift+F3" combo keys when starting up in Runtime from the editor.

## 3.5. Events

### Event Manager

The Event Objects allow commands or command lists to be executed in function with a project Tag state change or according to a time schedule that can be associated to the actual event.

The commands can therefore execute any project operation according to all the various possibilities that can be set as described in the topics relating to Object Commands. That is to say that an event can execute the same commands that can be executed by command buttons but the execution of these commands will be determined according to a time schedule or a Tag status.



The Commands on Event are very handy in projects that have to:

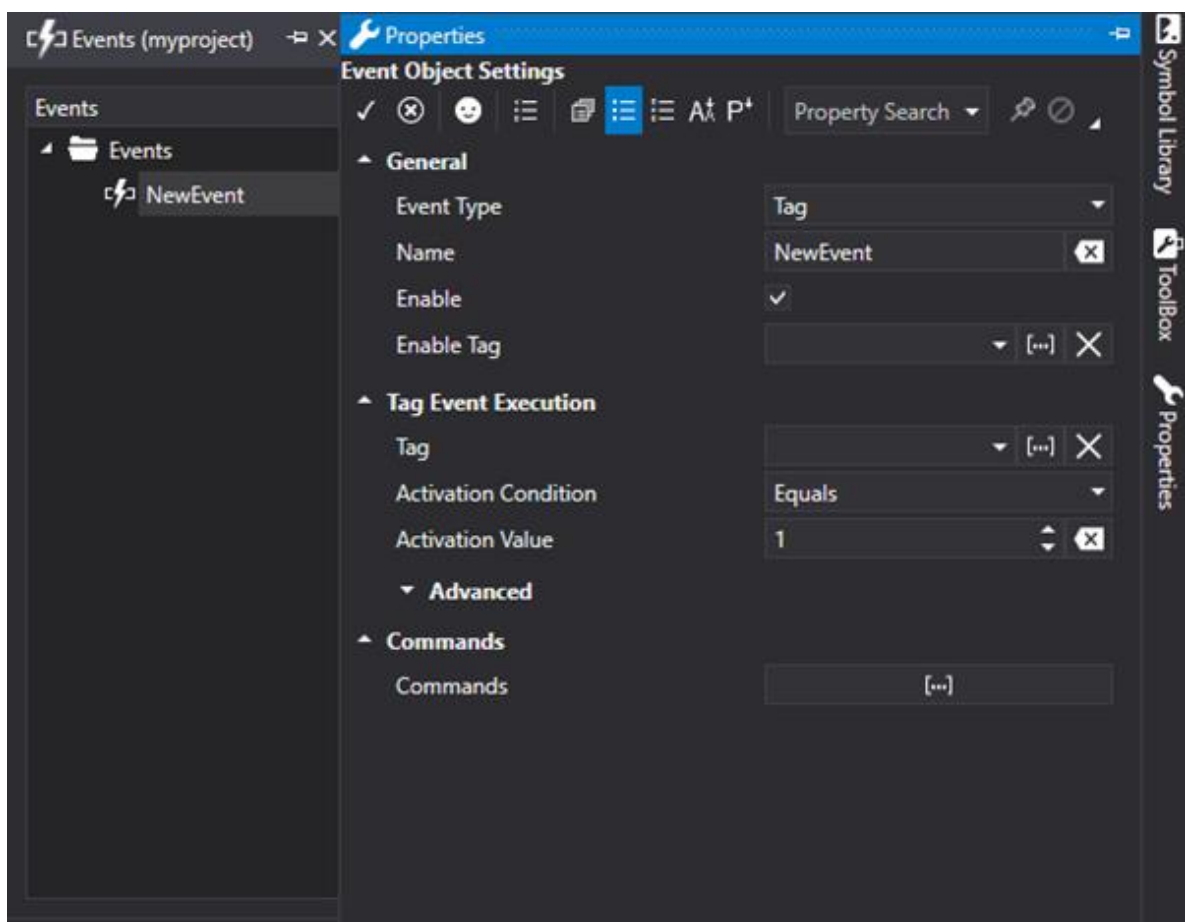
- manage operations for logic, commands, change pages, launching scripts, actions in recipes or reports for example, based on a Tag status change.
- manage operations or command as above but at certain prefixed time intervals (schedulers).

The possibility to manage Event objects saves the programmer a great deal of time in having to write code needed to execute one or several commands. However, the VB.NET script management can always be used for managing the more complex logic combinations.

## Commands on Event

To add Event objects to the project, you will need to select the "Event Manager" resource from the project's tree structure and then use the "Add New Event" command from the Events Toolbar, or right click to open the menu where the "Add New Event" command can also be found.

Once you have added a new Event to the project, you will also need to set its properties by using its properties window.configurazione.



*This screenshot shows the properties window for configuring the added new Event object.*

## Event object properties

Event objects that are inserted in the project need to be configured by means of using their property windows.

### Type

This is used to define the event's function type. There are two options to choose from:

- **Tag** : the event is determined by the state and condition of the tag associated to it.
- **Schedule** : the event is determined by a predefined schedule.

The Event is then configured according to which option has been chosen:

### Name

This is used to assign the Event object with a name.

### Enable

This is used to enable or disable the Event object.

### Enable Tag

This is used to assign a Tag which will enable the Event object's function in runtime.

## Event Execution

### Tag

If the **Tag** function type has been selected, you will need to define the tag whose condition will determine the event's activation in runtime according to the property settings described below.

### Condition and Activation Value

This allows you to select the Tag's logic condition which will determine the Event object's activation. As a consequence it may be necessary to establish the 'Activation Value', i.e. the prefixed numerical value with which to verify the activation condition against the value of the reference Tag.

The condition options are:

- **Equals**: the reference tag value becomes equal to the set value.
- **Greater Than**: the reference tag value becomes greater than the set value.
- **Less Than**: the reference tag value becomes less than the set value.
- **Greater Than or Equal**: the reference tag value becomes greater than or equal to the set value.
- **Less Than or Equal**: the reference tag becomes less than or equal to the set value.
- **Not Equal**: the reference tag is not the same as the set value.
- **On Change**: the reference tag changes in respect to the previous value.



The commands are executed only at the rising edge of the set condition.

### Activation Value

The event's numeric activation value based on the selected condition. This value is taken into consideration when the event's reference tag is numeric type.

### Activation String Value

The event's activation string value based on the selected condition. This value is taken into consideration when the event's reference tag is string type. The use of ID String is not supported in this field.

### Scheduling Type

If you have chose the Schedule option as the event's function type, you will need to define the event's time schedule to determine the event's activation.

The schedule options are cyclic as follows:

- **Every Minute**
- **Every Hour**
- **Every Day**
- **Every Sunday**
- **Every Monday**
- **Every Tuesday**
- **Every Wednesday**
- **Every Thursday**
- **Every Friday**
- **Every Saturday**
- **Every Month**
- **Every Year**

### Date and Time

Based on the type of scheduling set beforehand, you can also set a precise time and date to activate the event accordingly.

### Activation Tag Value

If a variable is entered in this field then the reference value for the activation of the event will be the value of that tag. In this way it is possible to make the comparison value dynamic in runtime for the activation of the event.

If this variable is inserted, any activation values of the properties described above "Activation Value" and "Activation String Value" will be ignored.

## Event object commands

After having set the event's activation modes you will need to define a list of commands to be executed in function with the event.

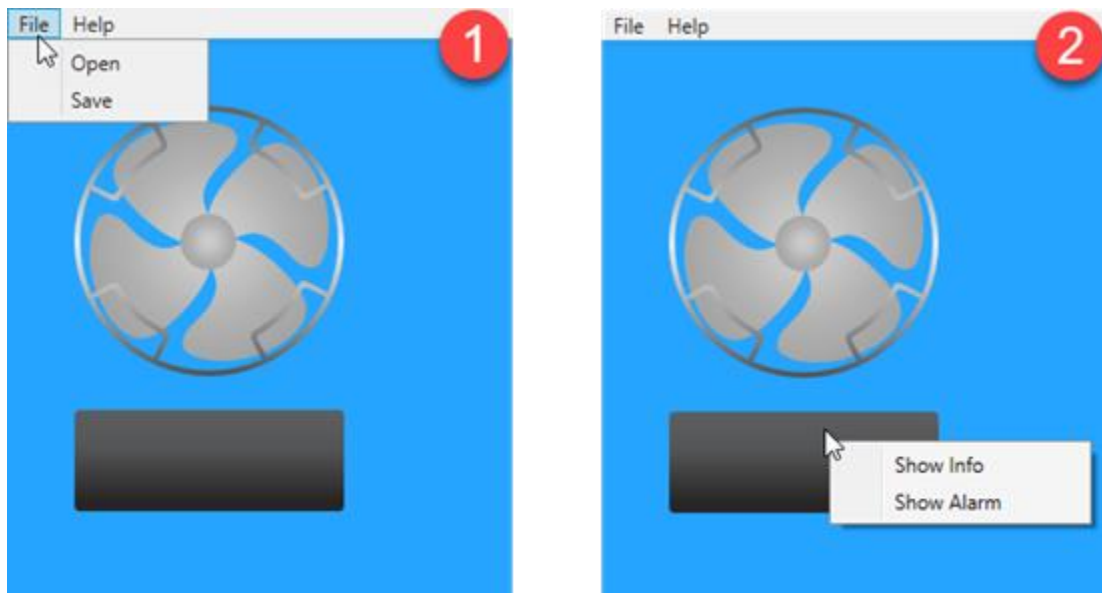
For details on commands that can be associated to a scheduler, please refer to the commands that can be associated to objects (e.g. command buttons).

## 3.6. Menu

### Introduction to using Menus

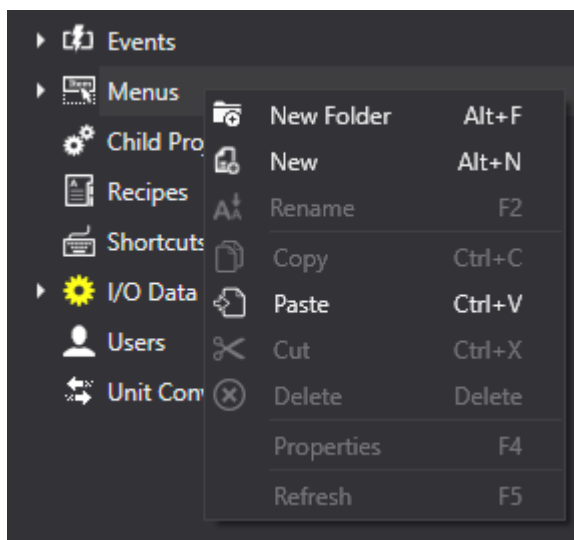
**Popup** menus can be created and activated with the mouse in Movicon. They can be organized in tree structures and with sub-menus or created as custom menu bars to associated to screen pages. The same commands that can be activated with buttons can also be activated through menus being those available in Movicon.NExT "**Command List**".





## Adding a new Menu

To create a new Menu, you can use the Project > Menu > Add New command or call the resource's contextual menu, as shown in the screenshot below, and through which it is also possible to rename it.



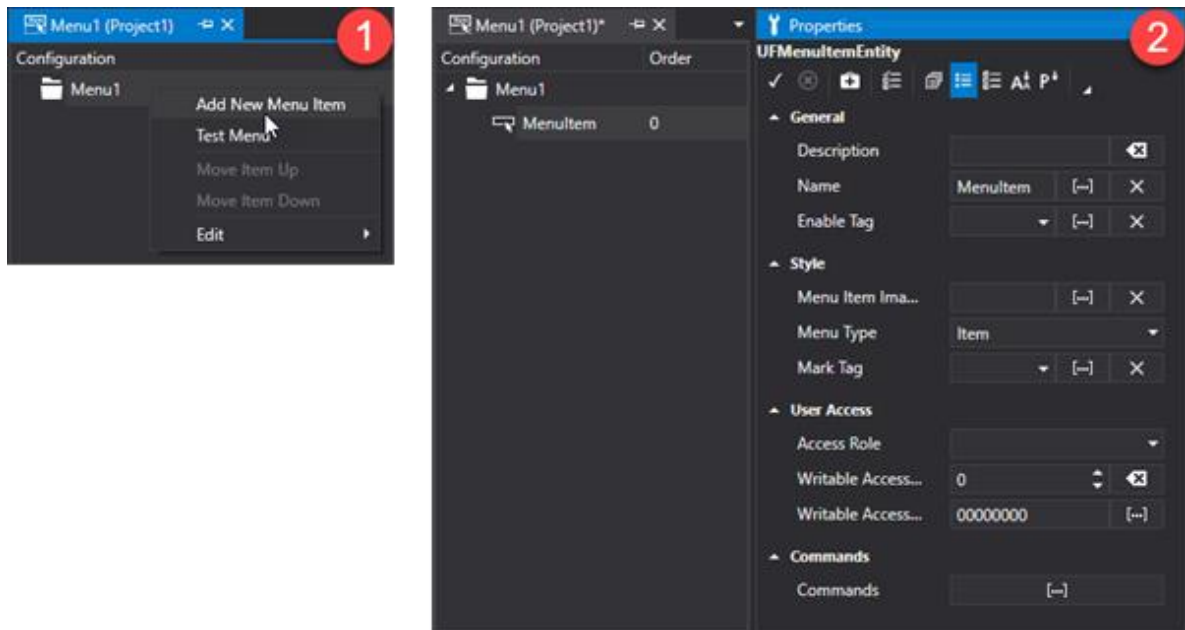
By adding a new menu it will automatically open in the work area. To open an already existing Menu double click on the resource in the Project Explorer window or use the icon / command contextual to the desired Menu resource.

When a new Menu is added, it will automatically open in the workspace. To open an already existing Menu simply double click on the resource in the Project Explorer window or use the desired Menu resource's contextual command icon. When double clicking on the Menu's name within the workspace, a dialog window will open enabling you to enter a description for the Menu. This description will not be shown in runtime.

## Inserting a Menu item

Each menu of the project can be composed of one or more items, or the actual components of the menu. The Items are displayed in the Menu tree in the work area, and it is possible to insert the Items that will

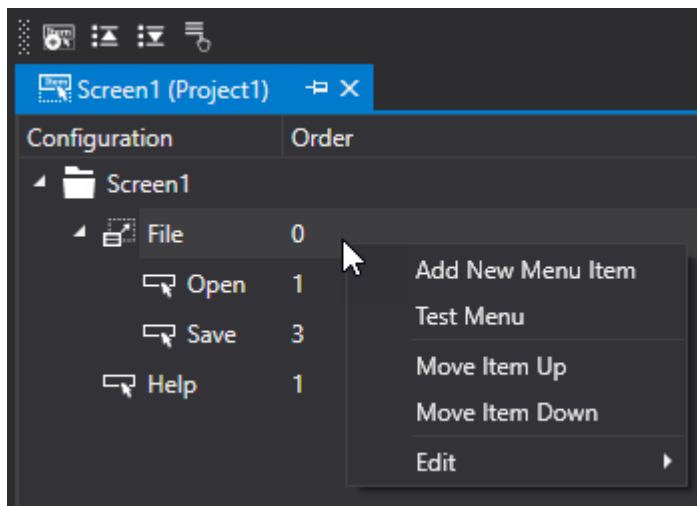
make up the menu, configure them and possibly structure them in sub-menus. Adding a new item using the right mouse key will automatically open a dialog window with the item's settings. The same window will also open by double clicking on the an already added item when needing to edit it.



*Window used for setting the Menu Item properties*

## Commands for editing Menu Items

Commands for editing Menu items are available from the toolbar or the contextual menu.



### Add New Menu Item

This command is used to insert a new Item in the Menu Resource being edited. Once the new item has been created a pop-up window will display containing the menu item's properties (described in previous paragraph).

### Move Item Up

This command is used to move the selected menu item one position up in the Item list of the menu being edited.

### Move Item Down

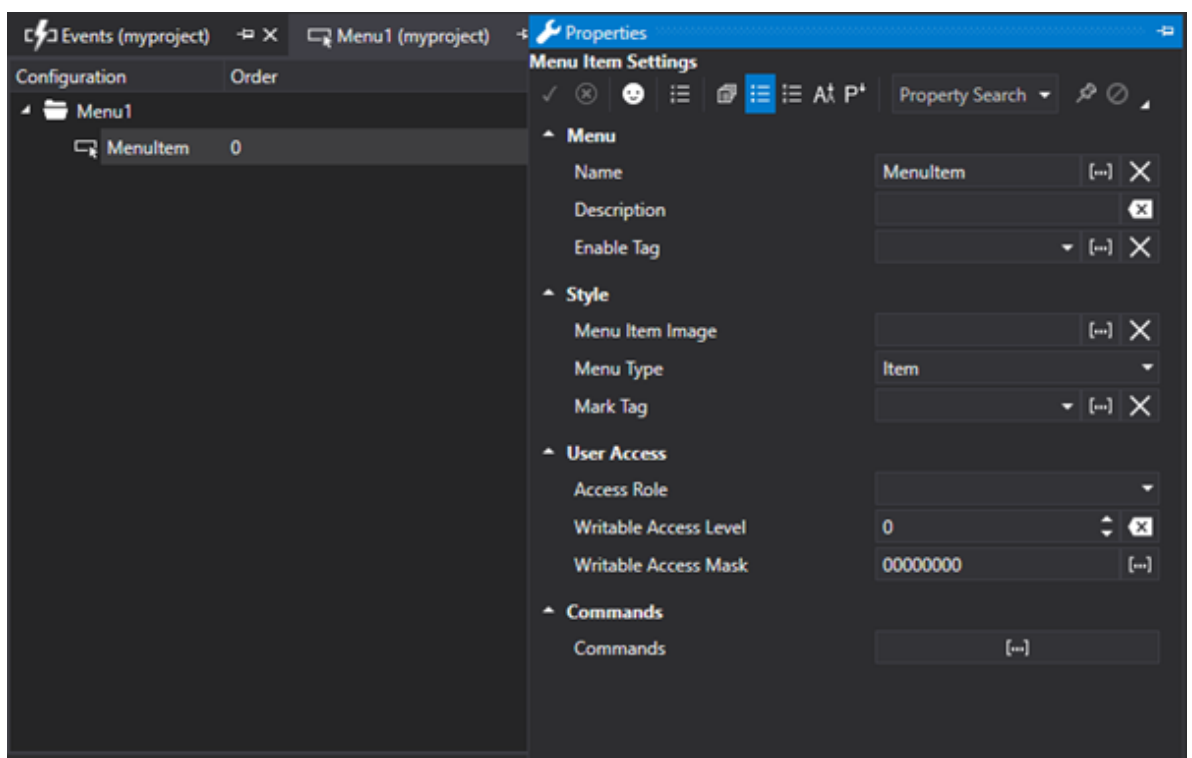
This command is used to move the selected menu item one position down in the Item list of the menu being edited.

### Test Menu

This is used to test how the menu being edited will display by showing an preview of the results obtained.

## Menu Properties

The Menu Item properties are:



### Name

Represents the name associated to the menu Item being edited. The menu Items support the Change Language function, therefore item names can be represented by fixed text or an "ID\_String" which must be declared in the string table. In this case the item name will change in runtime according to the language activated.

### Description

Represents a description associated to the menu item by the design engineer. This description will not be shown in runtime.

### Enable Tag

Allows you to select the variable that triggers the activation of "Menu Item". If the variable is reset to "zero", the menu item will appear inactive and therefore unavailable. Otherwise If the variable has a value different than "zero" the menu item will appear active and therefore available for executing the command. If you do not specify this option by leaving it blank, Movicon will consider the item as enabled.

## Style

### Image

Allows you to select an image to associate to the Item menu. The image default size is 32x32 px.

### Menu Type

Represents the item type to be inserted in the menu. This property can have the following values:

- **Item:** indicates that one or more commands can be associated to the item and can contain sub-items
- **Separator:** indicates that the item will be a simple separator between the menu items. It will be represented as a line.

### Mark Tag

This is used to select the name of the variable that will determine the display of the check mark (√) next to the menu Item. If the variable is set to a logic state different from "zero", the check mark appear will appear next to the item. If you do not specify this option, the check mark will not display.

## Commands

### Commands

This section is used for defining a list of one to several commands that will be executed when the menu item is activated. The editable Command List is the same one available for the Button objects.

## User Acces

### User Access Role

This is used to set the name of the user password group to allow user interaction with this object.

### Writable Access Level

This is used to set the object with a hierarchical user access level to allow user interaction with it as set by the User Password management.

### Writable Access Area

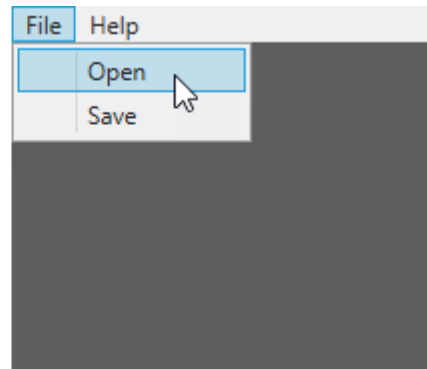
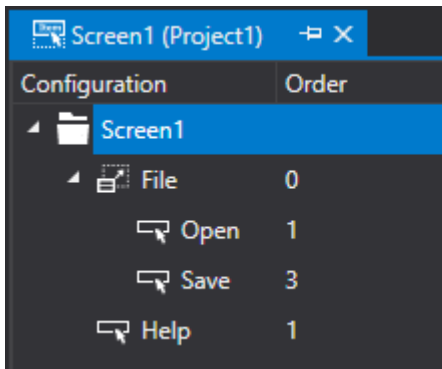
This is used to set the access mask in write for the object. If the connected user's access mask does not correspond to the one set in the object, they will not be able to change the tag value.

## Menu Activation in Runtime

The Menu resources can be activated only in screen menu bar format or as contextual pop-up menus associated to objects or symbols on screen in runtime.

## Screen Menu Bar

To activate a Menu as a customized Screen Menu Bar simply define the Menu with the same of the Screen you wish to activate. In this way the menu bar will appear at the stop of the Screen when it opens.



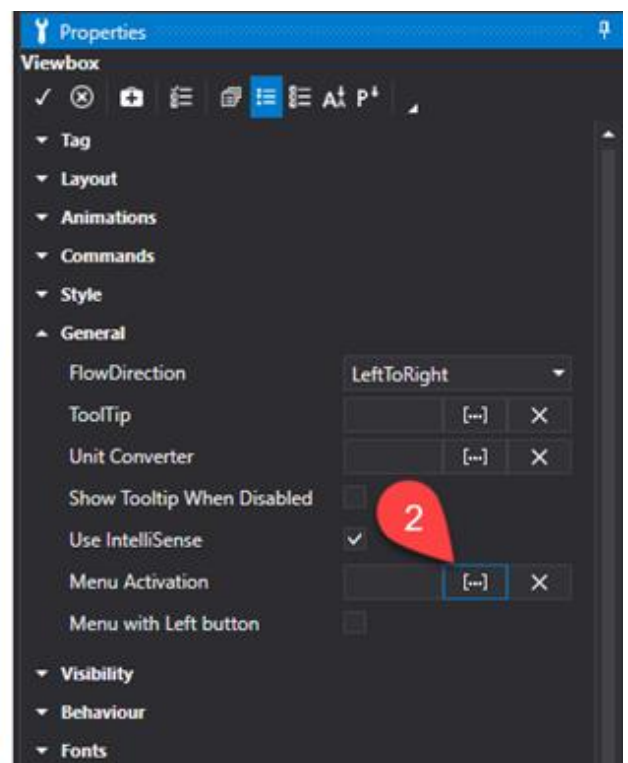
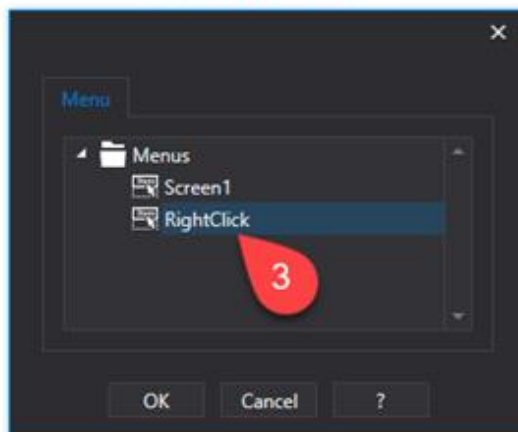
If the Screen has been inserted inside a folder (or in a nested folder), the Menu must be inserted inside a folder with the same name.



Defining a Menu the "Main" name, Movicon will activate for default this menu as a menu bar for all the Screens in the project that have not yet been defined with their own menu being a menu with the same name.

## Contextual Menus for symbols and objects

To activate a Menu as a contextual pop-up of an object or symbol, object's "MenuName" property will need to be set with the name of the menu to be displayed. In order to view this menu in Runtime simply right click on the object.



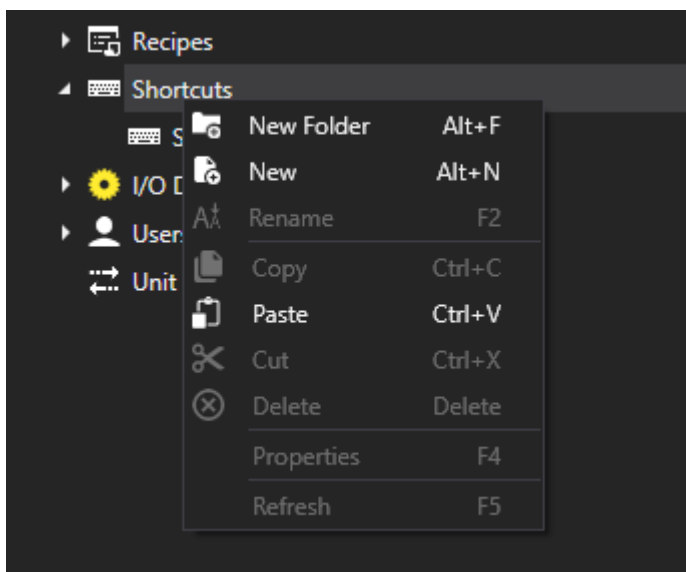
## 3.7. ShortCuts

### Introduction to using Project Shortcuts

When using Movicon you can execute a series of commands simply by using the keyboard. This is done by associating the command to keys or combination keys as shortcuts to commands in function with the screen currently active at that moment. These shortcuts can be used for activating the identical command that can also be activated with those buttons available in the Movicon "Command List". "Commands List"

#### Inserting new shortcuts

In order to create a new shortcut use the Project > Shortcuts > Add New command or call the resource's contextual menu as shown in the screenshot below (through which it is also possible to rename it).

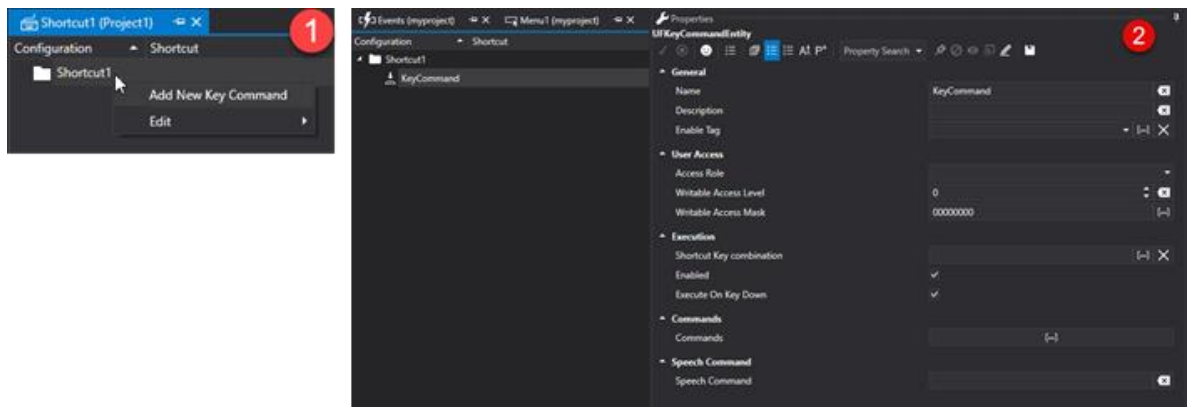


*Window used for adding a Shortcut Description*

When a new shortcut is added it will automatically open in the workspace. To open an already existing shortcut simply double click on the resource within the Project Explorer window or use the desired shortcut resource command icon. The new shortcut can be given a description using the dialog window which opens by double clicking on the shortcut's name within the workspace. This description will not show in runtime.

#### Inserting shortcut command keys

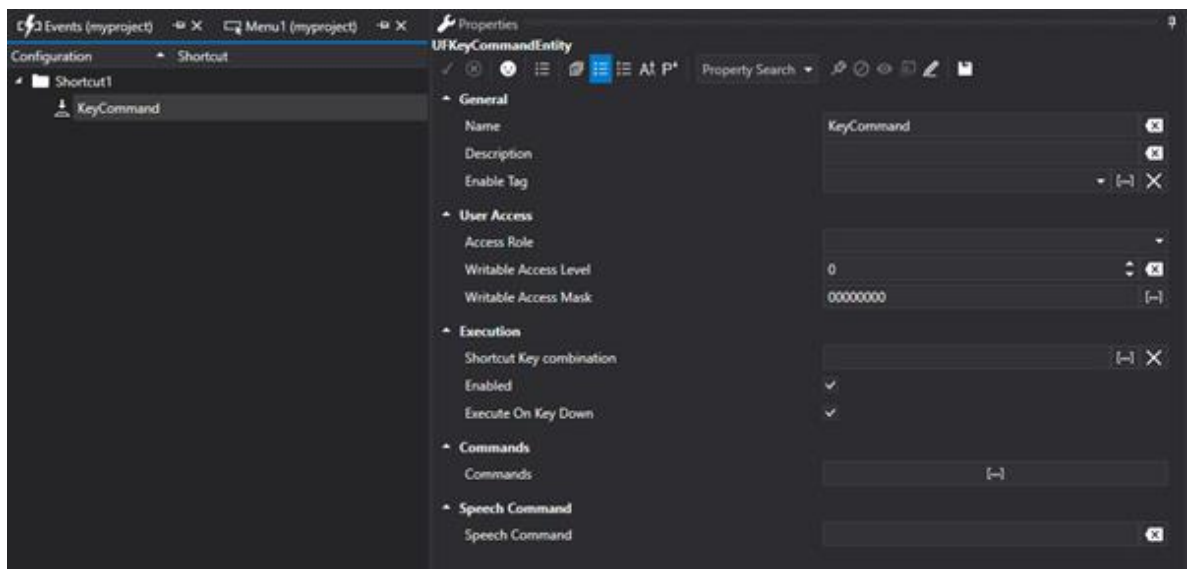
Each of the project's shortcuts can be composed of one or more Key Commands. The Key Commands are displayed in the tree structure of the Shortcut displayed in the workspace where it is possible to insert the keys and configure them using the appropriate commands. When adding a new Key Command a dialog window will automatically open displaying its settings. This window can also be opened with a double click on an existing key to modify its settings.



*A Shortcut Key Command property setting window*

## Shortcut Properties

The properties of a Shortcut key command are as follows:



### Name

Represents the name associated to the Key Command. A shortcut can actually contain a list of several Key Commands.

### Description

Represents an eventual description that can be associated to the Shortcut key command by the programmer. This description will not be shown in runtime.

### Shortcut

Represents the combination keys (or key) requested for activating the command, or command list. This property is used for specifying the key to be used by clicking in the text box and using the keyboard to enter it manually. As an alternative you can also use the "..." button to the right that once clicked will open a pop-up window where the desired combo keys to be associated to the shortcut can be entered.

### Speech Command

specifies the "Voice Command", to be used at runtime, to execute the accelerator.

### Enabled

Enable / disable the accelerator in development.

### Enable TAG

Allows you to select the variable that triggers the activation of "Menu Item". If the variable is reset to "zero", the menu item will appear inactive and therefore unavailable. Otherwise If the variable has a value different than "zero" the menu item will appear active and therefore available to the execution of the command.

If you do not specify this option, leaving the option empty, Movicon will consider the item always enabled.

### On Key Down

This property indicates whether the command associated to the created combo keys is to be executed when the key is pressed down (key Down) or released (Key Up).

#### 3.7.1. Shortcut Activation in Runtime

The shortcut resources can be activated directly from the screen containing them in runtime. To activate the shortcuts of a specific Screen simply enter the Shortcut with the same name of the screen you wish to activate. In this way when the Screen opens, the corresponding Shortcut will activate to allow command execution upon pressing its corresponding keyboard keys.



If the Screen has been inserted in a folder (or in a nested folder), the Shortcut must be inserted in the folder with the same name.



When defining a Shortcut with the name of "Main", Movicon will activate this Shortcut for default for all the project Screens that have not been defined with their own Shortcut, that is one with the same screen name.



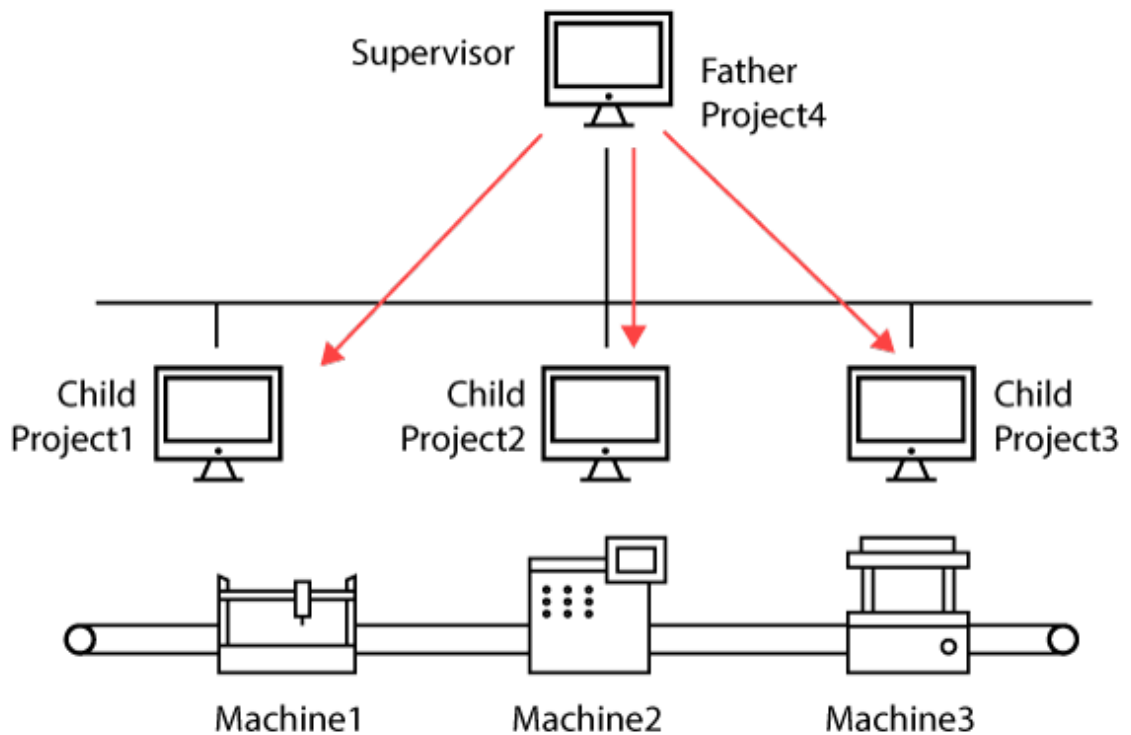
When wishing to activate commands via the keyboard for a georeferenced screen, you will need to create a keyboard command named "Geopage"

## 3.8. Child Projects

Movicon.NExT is used for structuring projects by decentralizing the resources to other projects (child) with dynamic relationships to enable project distribution.

Movicon.NExT offers a powerful and innovative feature to confront the new challenges in design engineering supervision systems. Child projects are normal Movicon.NExT projects that although designed to function autonomously can be linked to parent projects to establish "Parent-Child" relationships due to which enables the Parent project to provide all Child project resources as if they were its own.





A project can be associated with several Child Projects. Likewise the Child project can also be a Parent project to several other Child projects thus creating a drop-down tree structure of projects.

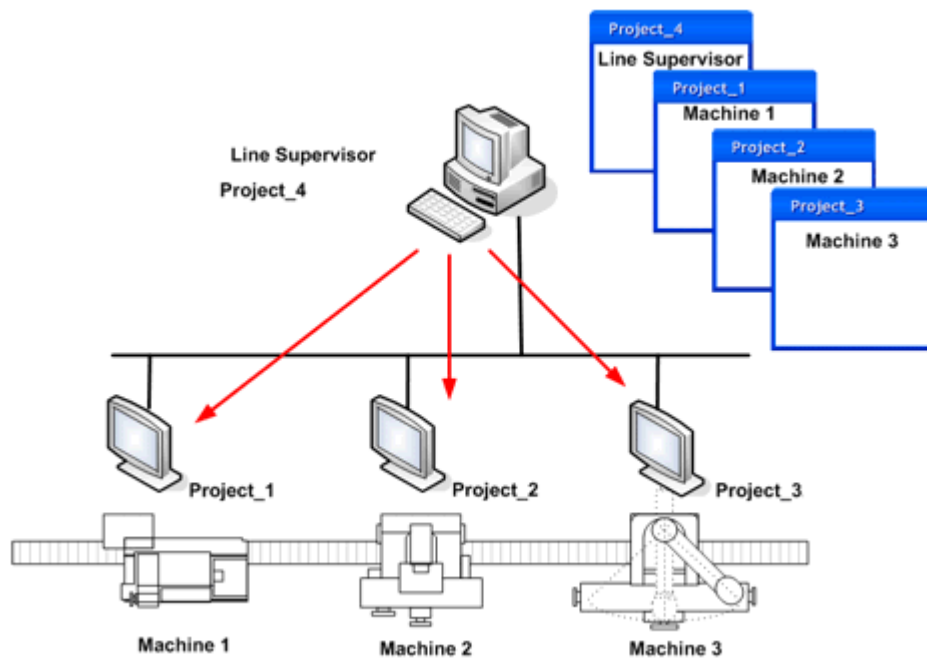
This possibility opens the way towards many types of advantages. The main ones are:

## Distributed Project Design Engineering

Project structured with Parent-Child relationships propose many advantages to those companies that rely on team work. In respect to those traditional technologies where several people work on the same project at once sharing out tasks, Movicon.NExT instead offers the possibility to distribute such tasks over diverse and independent projects. This allows the Team Leader to be in charge of team members through his Parent project by using their Child projects which at the same time are completely independent. The relationship between projects is established through Dynamic Linking and not englobing and therefore the Parent Project will be provided with all automatically updated Child project resources to enable team members to continue updating their projects autonomously. The Parent Project will be able to contain all the Child project resources without having to distinguish between resource names or duplicate resource names because the name difference will be given by the child project path. Therefore, a VAR0001 for example can coexist in the Father project as well as in a Child project by being identified by their name and project path.



Note: This architecture provides the option to automatically startup child projects in runtime when the Parent project is started up in runtime. This option is called "Child Project Setting Options" property and is found in the Project properties.



## Distributed Execution

The use of Parent-Child project relations is very handy for modular plant systems or machines where the plant is divided into different areas that are also independent from each other. In this case diverse projects can be created, one for each area, and then integrated into one unique Parent Project from which pages and variables of Child Projects can be accessed.

### Example:

An automated production line composed of 3 independent machines: Each machine has its own project run locally on the machine's PLC. The machines are then integrated in the production line and connected to the General Supervisory PC.

The great advantages offered by Movicon.NExT's capability to drastically save development time is to realize general supervision projects such as a "Parent Project" with the three machines' "Child Projects" residing in the Local PCs.

In this way, the Parent project can automatically get all the individual variables of the various child projects, to produce general summary layout screens. Screens from each of machines can then be accessed from the general layout screens residing in the Parent Project by simply opening the child project screens residing in the machines' local PCs. Not only does this offer the advantage of reducing project development time but also the advantage of having all machine modifications automatically available in the general supervisor as well.



Note: In the example using this architecture, child projects do not need to be preset to startup automatically in runtime when the Parent Project is started up in runtime (being autonomous projects) by enabling the option provided in the Project's "Child Project Setting Options" properties. However the child projects must be defined with the Server (supervisor) project's IP address in these settings.

## Child Project Paths

The "**Child Projects**" can be retrieved by using the PC's local path where the "Parent Project" resides and a network path.

In cases where the project is local, it would be best to insert the folder containing the "Child Project" in the "Parent Project" folder, despite the fact that any path can be defined. However by doing this the "Parent Project's" search path will always relate to the "Parent Project's" path and problems finding absolute paths can be avoided when moving the entire "Parent Project" to a different path or another PC.

If the "Child Project" resides in the "My Documents" folder, it will always be searched for in relation to the "My Documents" folder of the user logged on to the operating system at that current moment.

If the "Child Project" is in another folder or in a network computer, the search path will be a physical search path. Therefore in this case it must be made explicit that the network PC folder must be shared.



It is always more convenient to insert the "Child Project" in the "Parent Project's" folder or subfolder whenever possible to avoid using absolute search paths.

## Accessing Child Project Resources

The great advantage of using "Child Projects" is that the resources of both "Parent Project" and "Child Project" are easily accessible to both projects. For example, "Parent Projects" are able to call "Child Project" screens and vice-versa.

In fact during Runtime mode, screen pages from both projects can be displayed without it being obvious that they come from different projects.

Parent and Child Project access is established by simply using a slightly different syntax during project design mode. When using the browser window to select resources on the Parent Project side, it is normally possible to see the Child Project resources as well. This will facilitate the selection of the resource desired. However, this is not possible on the "Child Project" side and, therefore, requires that the name of the desired resource be entered manually.

The syntax used for accessing the "Child Project" resources from "Parent Project" side is:

ChildProjectName\ResourceName

for example:

ChildProject\Screen 1  
ChildProject\VAR00001  
ChildProject\Basic Script 1  
ChildProject\Menu 1

The syntax used for accessing the "Parent Project" resources from "Child Project" side is:

..\ResourceName

for example:

..\Screen 1  
..\VAR00001  
..\Basic Script 1  
..\Menu 1

The different levels of nesting must also be taken into consideration. For example, in situations where the "Parent Project" has two child projects: "Child Project1" and "Child Project2", the syntax below must be used to access the "Child Project2" resources from the "Child Project1" side:

..\ProjectName\ResourceName

for example:

..\Child Project2\Screen 1  
..\Child Project2\VAR00001  
..\Child Project2\Basic Script 1  
..\Child Project2\Menu 1

Another situation would be that a "Parent Project" has a "Child Project" which has its own "Child Project". Therefore, this would result as having "Parent Project", a "Child Project" and a "Child Project2". The syntax to use for accessing the "Child Project2" resources from the "Child Project" in this case will be:

ChildProjectName\ChildProject2Name\ResourceName

for example:

ChildProject\ChildProject2\Screen 1  
ChildProject\ChildProject2\VAR00001  
ChildProject\ChildProject2\Basic Script 1  
ChildProject\ChildProject2\Menu 1

The syntax to use for accessing the "Parent Project" resources from the "Child Project2" side is:

..\..\ResourceName

for example:

..\..\Screen 1  
..\..\VAR00001  
..\..\Basic Script 1  
..\..\Menu 1



Caution! If local screen variables have been defined for the screen being used, the syntax to use for accessing the parent project variables may be different. If the parent project variable has the same name of the local variable, the following syntax should be used:

..\..\<variable name>

The "..\" suffix is used for returning back from pointing to a hierarchy of variables. The hierarchy is as follows:

Local Screen Variables -> project Variables -> parent project variables

It is also possible to use the variables between Parent and Child in the objects' basic expressions. However, it is important to take into consideration those local variables that are not supported in basic

scripts. In this case, when you insert a basic expression in an object in the Child Project, the syntax to use is:

[VAR00001] + [VAR00002] -> the Child project variables are loaded  
[..\VAR00001] + [..\VAR00002] -> the Parent project variables are loaded  
[..\\.\VAR00001] + [..\\.\VAR00002] -> no variables are loaded due to reference error

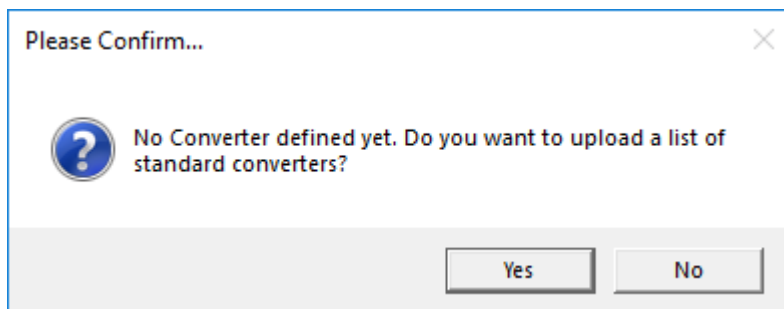
## 3.9. Unit Converter

This resource is used to convert numeric values from one unit of measure to another by applying the corresponding conversion factors.

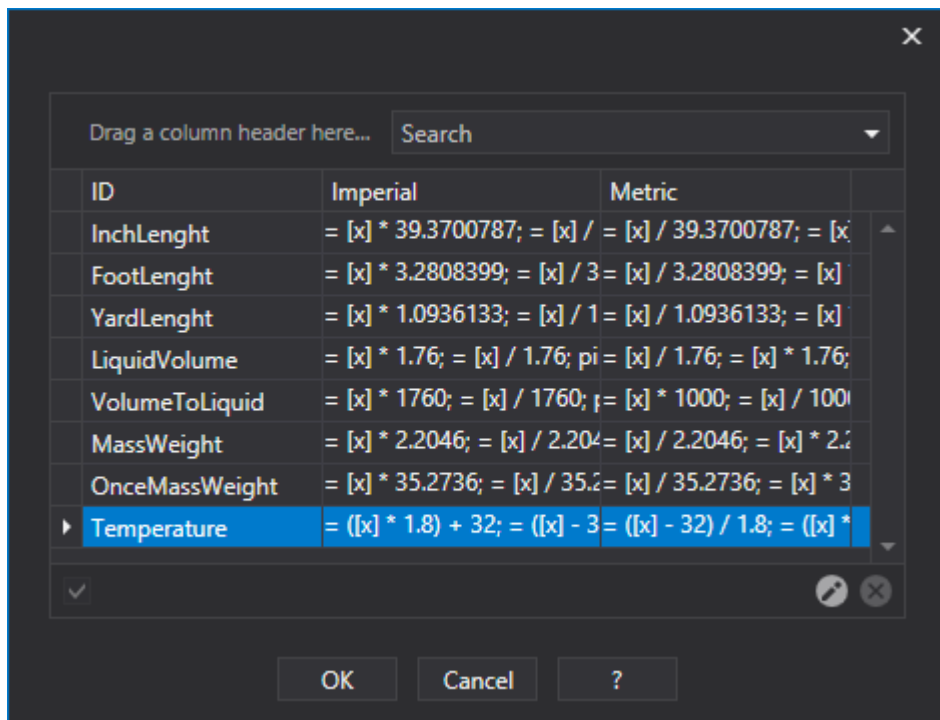
Some common conversion examples are:

- metres/inches --> 1 metre = 39,3701 inches --> Expression = [x] \* 39,3704
- °C/°F Degrees --> 1 °C = 33,8 °F --> Expression = [x]\*1,8+32

When opening the Unit Converter resource for the first time, you will be prompted whether load a list of standard converter or not.



When confirming with yes, you will be shown a selection of converters that can be imported into the project:



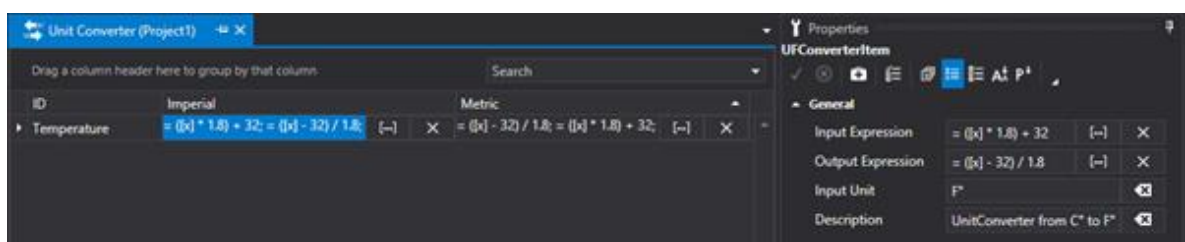
Press OK to import the predefined converters for Metric and Imperial Measurement Systems.



You can always add other Measurement systems using the contextual menu's "Add new Conversion System" item.

In the following screenshot shows a Temperature unit measure converter for the two Metric and Imperial Systems.

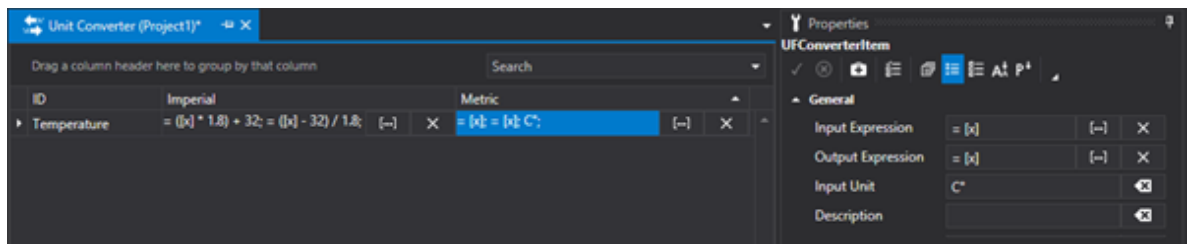
The "Imperial" column reports the expressions (in read and write) that will be applied to objects when setting the Unit Converter to the Imperial system at runtime. The "Metric" column reports the expression (in read and write) which will be applied to objects when setting the Unit Converter to the Metric system at runtime.



The presence of the [x] syntax in the conversion expression refers to the tag associated to the graphical object to which the converter is applied.

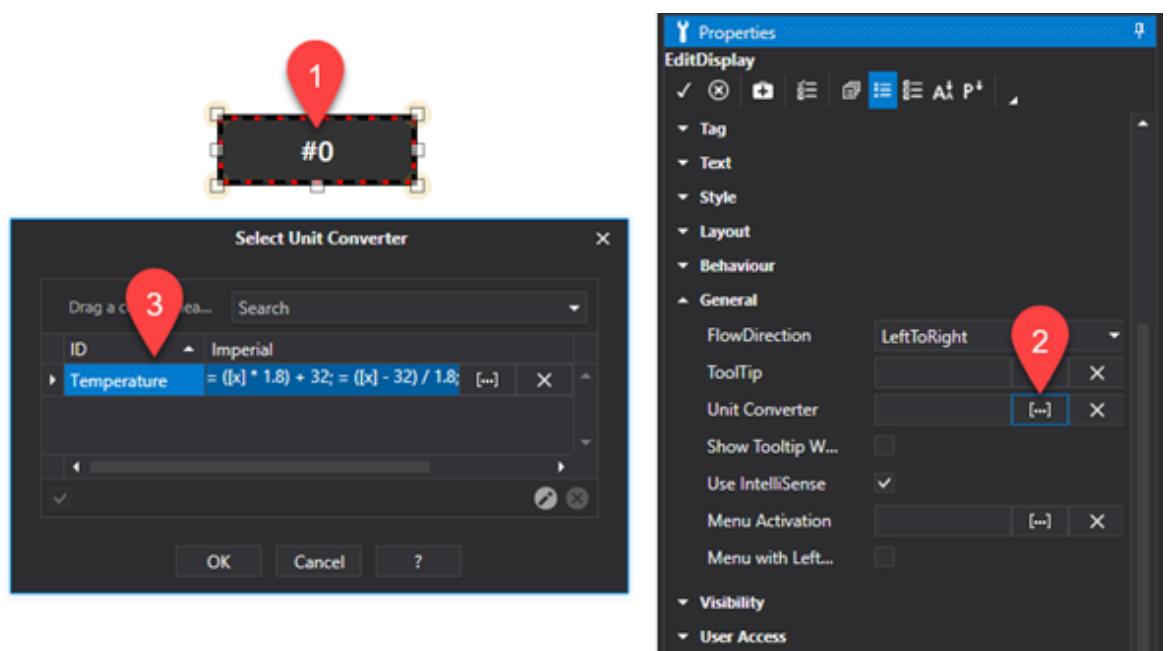


If the data shown is connected to a Tag read from a device based on the Metric system, it will not be necessary to apply any conversion when setting the Metric unit Converter at runtime. In this case, the input and output expressions will be "= [x]" with no need to apply any transformation.

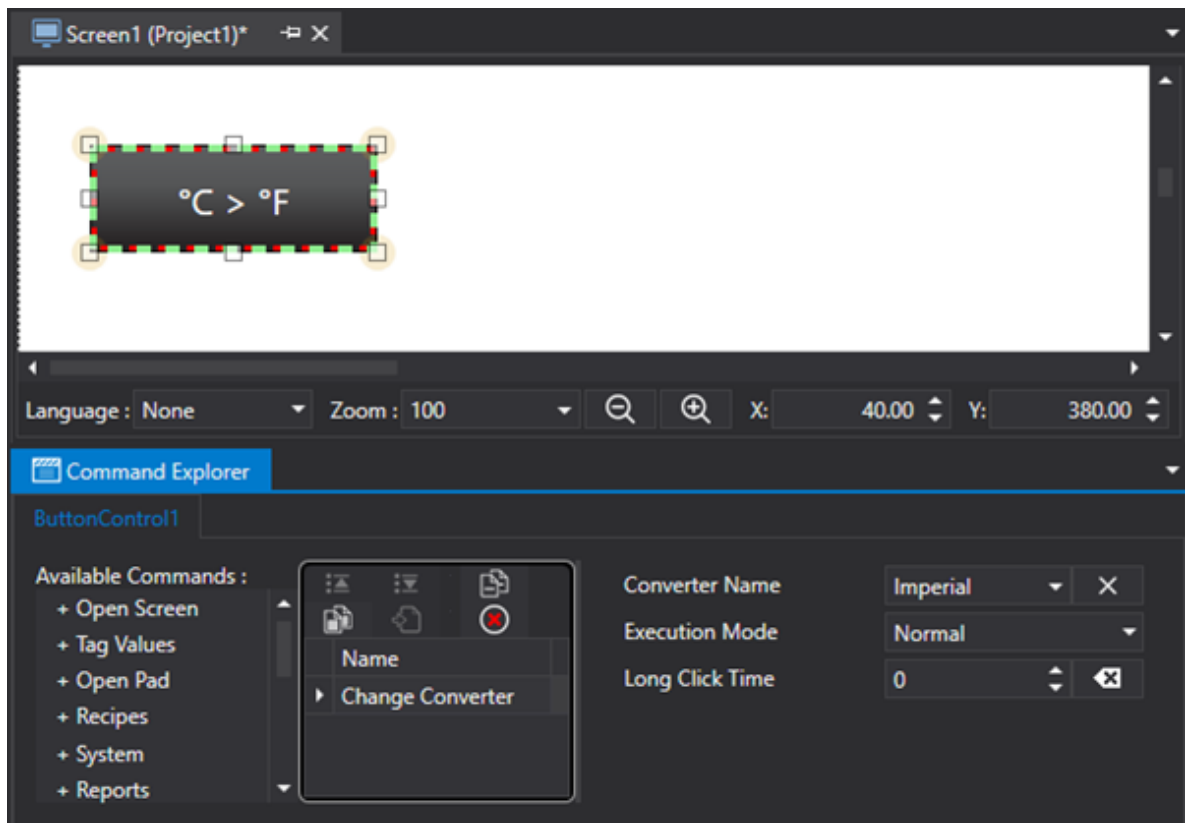


Once the different Measure Units in rows and relative Conversion Systems in columns have been configured, it will be possible to:

1. Insert the converter in the properties of the graphical object that must apply the conversion when displaying numeric values (such as the display object).



2. Predispose a button to associate the "Change Converter" command to in order to allow users to activate the desired Conversion System.



The presence of a Converter in a graphical object can be associated to that of an Expression. However, be careful because the Expression is evaluated after the converter has been applied while being displayed. .

First we will add a "New Conversion System" and then add all the desired converters within it.

## 3.10. Speech Recognition

Speech Recognition can be used in Movicon.NEXT project to execute a series of commands using voice controls. This results particularly useful in all those situations where operators need to keep their hands free and invoke commands quickly without the aid of a console.



In order to use the Speech Recognition, you will first of all need to install the speech recognition engines using this link: <https://www.microsoft.com/en-us/download/details.aspx?id=27224>

The speech command management in Movicon.NEXT is found in some properties belonging to the project and the object you wish to use for executing the speech command.

## Project Properties

### Speech Culture

This is the language with which the speech recognition engine interprets the invoked speech commands. Obviously, the corresponding speech recognition engine should be installed.

### Default Speech Commands

This is a list of 8 default commands that execute in order of:



## Client Desktop

- next, navigates to next screen according to how the resources appear in the tree.
- back, navigates to the previous screen according to how the resources appear in the tree.
- home, navigates to the home screen when project is started up as MainScreen, or the TilePage when the project is started up in TilePage mode.
- reset, resets zoom command effects
- swipe up, simulates swipe up movement used in touchscreens
- swipe down, simulates swipe down movement used in touchscreens
- zoom in, enlarges screen size
- zoom out, reduces screen size

## App

- next, navigates to the next screen based on the browsing chronology
- back, navigates to previous screen based on the browsing chronology
- home, navigates to the Movicon.NExT Web Client home screen
- reset, resets zoom command effects
- swipe up, has no effect as app bars are missing
- swipe down, has no effect as app bars are missing
- zoom in, enlarges screen size
- zoom out, reduces screen size

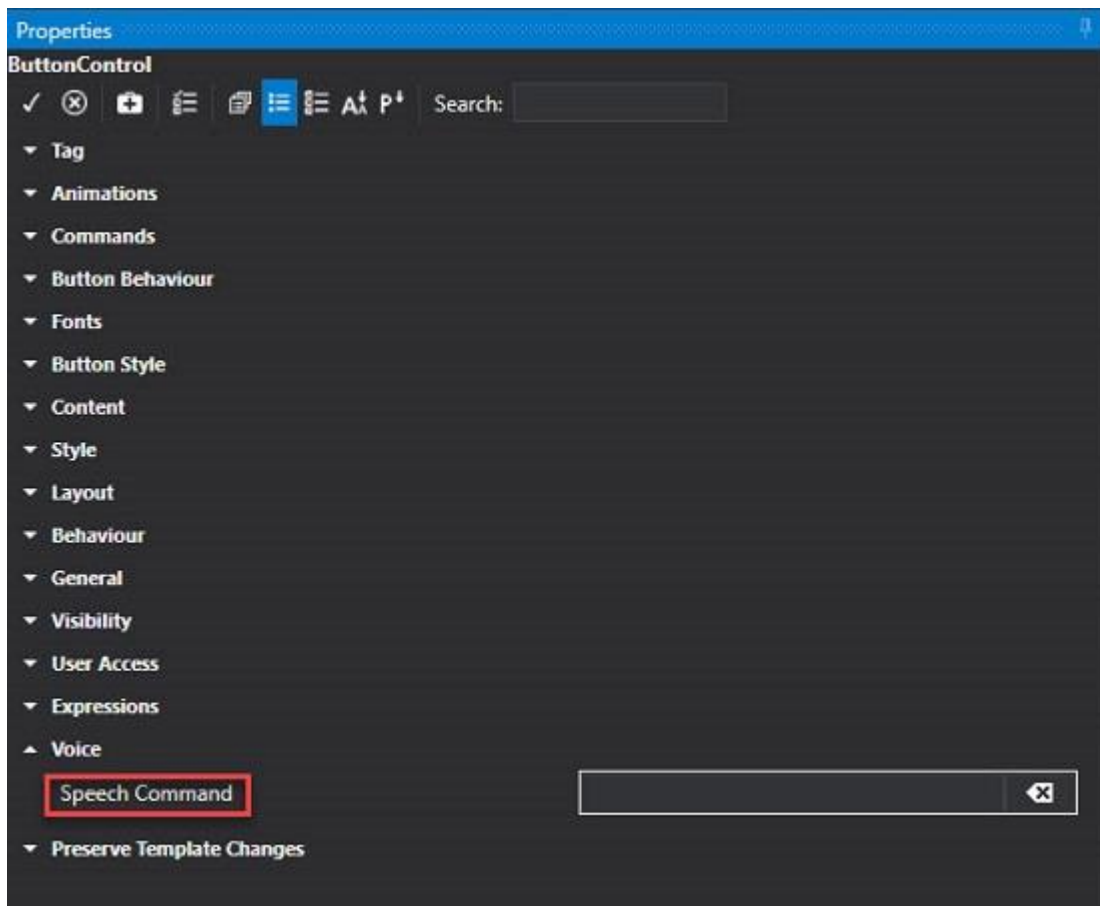
## Enabling Commands at Runtime

To activate speech recognition at runtime, you will need to enable the relevant flag in the Voice group of the project's properties.

The Speech Confidence level, (values between 0 and 1) represents the threshold below which the command will not be acknowledged, for which a higher value increases the interpretation permission of the command, meaning that the system will become more selective, and viceversa.

## On Screen Object's Speech Commands

As described before, by using speech commands in objects, commands associated to the object can be executed. To configure this resource, you will need to associated one or more commands to the object and then specify in the "Speech Command" property the term used to activated the desired commands. For example, the term "Start" could be entered in the "Speech Command" property:



By pronouncing the insert term "start", the list of commands previously set in the object will be executed.



A microphone and an audio are essential to the correct functioning of the Speech Commands. English is the only language currently supported for the terminology used within the speech command ambient.

### Change Language

When changing language at runtime, the system will attempt to use the right speech recognition engine, if present, otherwise it will use the default one. When commands are present in the project's string table, the system will use translation to impart the correct command. Let's look at example of this:

- button associated with the "on" speech command
- String table with "en-US", "it-IT"
- The "on" string corresponds to the "accesso" string in the "it-IT" string table.
- When launching runtime and selecting the "en-US" language, the system interprets and executes the "on" speech command.
- When launching the runtime and selecting the "it-IT" language, the system interprets and executes the "accesso" speech command.

## 3.11. Database Configuration Tool

The "SQL Database Configuration Tool" is used to create tables deriving from the main DataLogger table with average, minim, maximum data value aggregations based on minutes, hours and days. Basically, each table contains a record for each aggregation time interval (minutes, hours, days) and each column

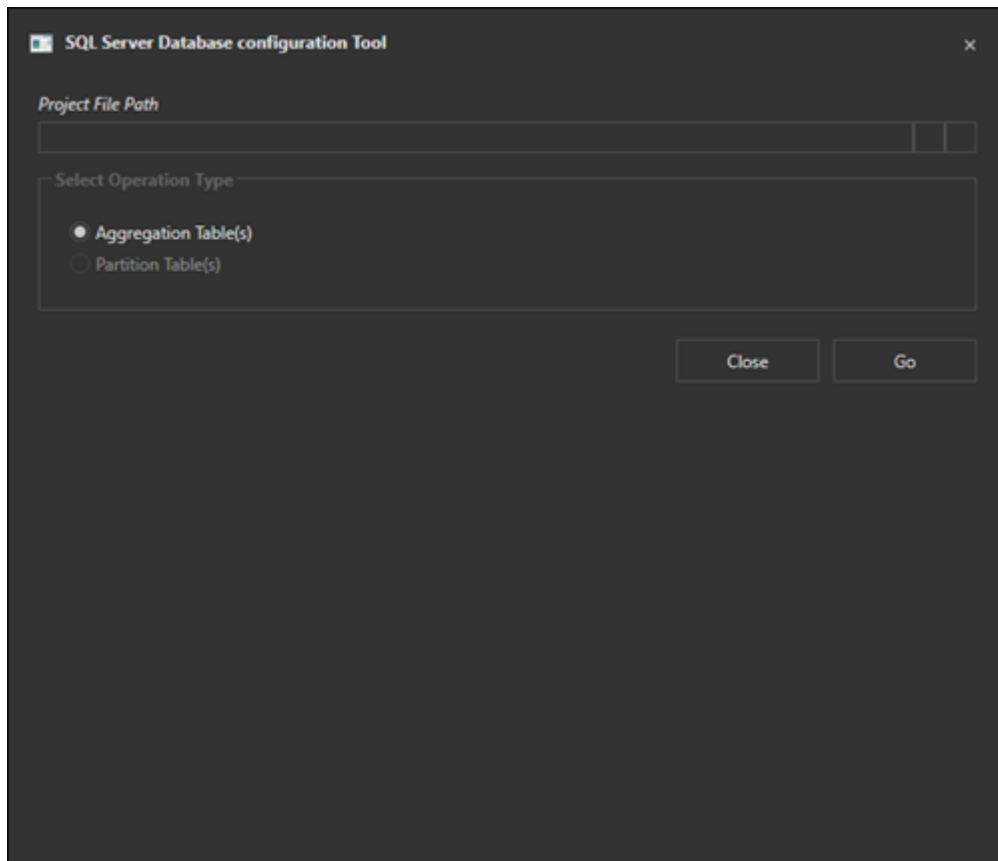
from the DataLogger of origin will generate 3 columns, one with the average value, one with the minimum value and one with the maximum reference value.

### DataLogger table aggregation

Aggregating Datalogger tables is basically done in two ways:

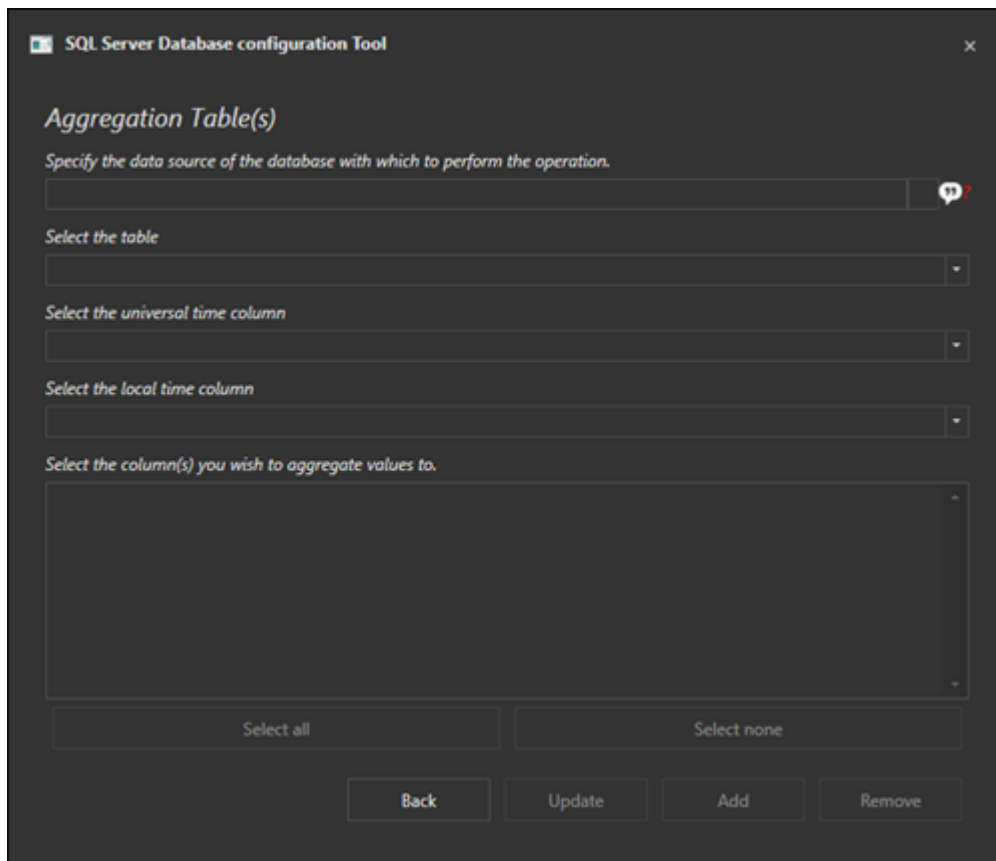
#### Using the User Interface

Launch the C:\Program Files\Progea\Movicon.NExT x.y\SQLDatabaseConfiguration.exe executable with a double click to open this interface:



Select a project from the first parameter and press the Go button to start up the procedure to aggregate tables of DataLoggers configured in the project.

Table aggregation can also be done without selecting a project by using the Go button as shown in the screen shot below:



- **Data Source:** use to specify the database in which the DataLoggers to be aggregated are.
- **Table:** use to specify the DataLogger table to aggregate.
- **Universal Time Column:** use to enter the name to the column containing the UTC data.
- **Local Time Column:** use to enter the name to the column containing Local Time data.
- **Select columns:** use to select the columns, numerically, to aggregate.

Once all these settings have been completed, the '**Update**' and '**Add**' buttons will activate to allow existing aggregation updates to add aggregated tables respectively, along with the necessary storage procedures and triggers.

If the selected table has already been subjected to aggregation beforehand, the Update button will also be enabled.

### Using the command line

The tool in question can also be called using the command line with the following options:

- /S:** silent execution
- /F:** NExT project path
- /W:** project protection password if any
- /O:** operation type to be performed: "1" to aggregate tables

Example of command line parameters:

```
/F"E:\MyDownloads\TestDatalogger1\TestDatalogger1\TestDatalogger1.UFProject" /W"mypassword" /O"1" /S
```

Other advanced options which reflect the description of the user interface already seen above:

- /P:** command type: "A" (Add), "R" (Remove), "U" (Update), "C" (Check)
- /D:** database connection string, in ADO.NET or DevExpress format
- /T:** table name

**/U:** UTC column name  
**/L:** Local time column name  
**/D:** Data column names to be aggregated  
**/H:** Hide column names to exclude them from the interface selection list.

## 3.12. Deploying Projects without WebHMI

As from version Movicon 4.1 you will be able to use the "Deploy Project" tool to transfer your Movicon.NExT project from the development machine directly to the predefined path in the target machine.

The same Movicon.NExT and corresponding build must be installed in both the development and target machines.

The Deploy Server will be installed and started up automatically with the Movicon.NExT software installation.

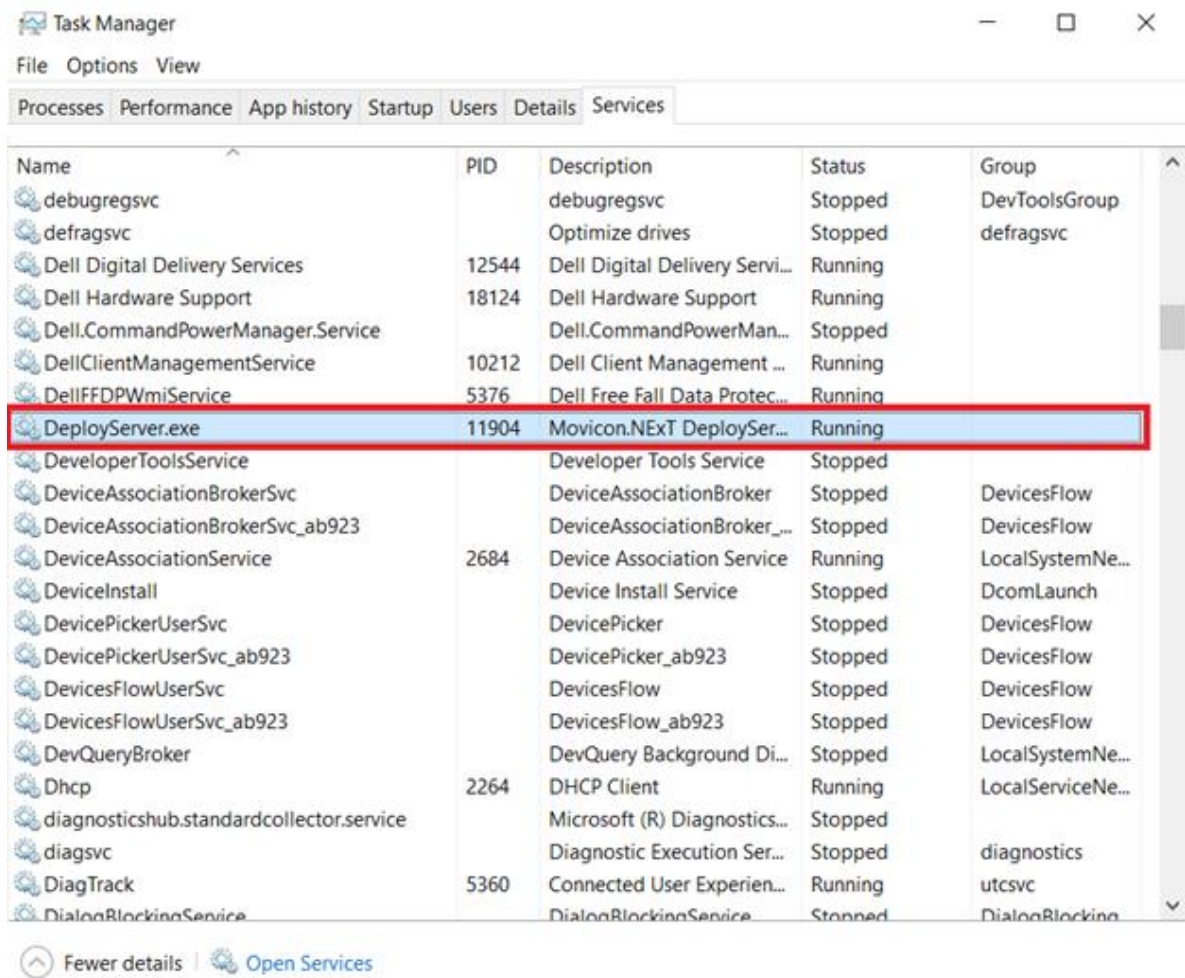
To transfer the project to the target machine proceed as follows:

1. Install Movicon.NExT 4.1 on the target machine (the same version and build installed in the development machine).
2. Edit the target machine's "appsettings.json" file in the "C:\Programmi\Progea\Movicon.NExT 4.1\DeployServer-Files\" folder by defining the access credentials (User, Password) and the destination path folder for the Deploy Server.



```
[
  "RaygunSettings": {
    "ApiKey": "3bydG2PZfwZHixwL8akyyQ"
  },
  "Login": {
    "User": "myUser@myDomain.com",
    "Password": "Password123$"
  },
  "DeployServerSettings": {
    "Path": "C:\\MyDestinationPath\\",
    "Hosted": false,
    "ProxyPath": "",
    "WebHMIProxyPath": "",
    "MaximumMessageSize": 2097152,
    "ListeningPort": 5002,
    "ProcessOutputMaxLength": 5000
  }
]
```

3. Restart the "DeployServer.exe" service using the task manager or restart the target machine to force the re-initialization of the Deploy Server with the new setting defined in the "appsetting.json" file (User, Password, Path)



4. Open Movicon.NExT on the development machine and open the "Deploy Project" window ("Options" Menu -> "Deploy Project").
5. Enter the access credentials previously defined in "appsettings.json" in the target machine.

Deploy Project

▲ Profile Info

Profiles NewProfile New Save Delete

Profile Name NewProfile

User myUser@myDomain.com

Password .....

Host https:// 172.16.125.37:5002

Reconnection Timeout 0

Overwrite Retentive Files ☐

▲ Commands

Connect Disconnect

**Target Machine: Control Commands**

Deploy Server and Project

Delete remote project

Start Servers

▼ Advanced Servers Start

▼ Remote License Info

▲ Upload Info

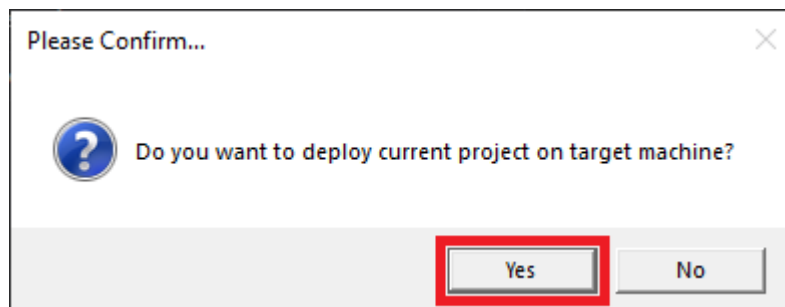
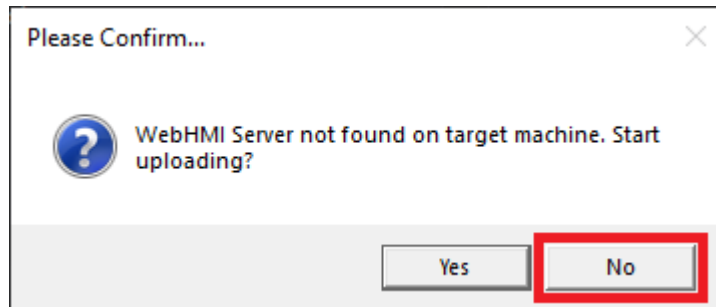
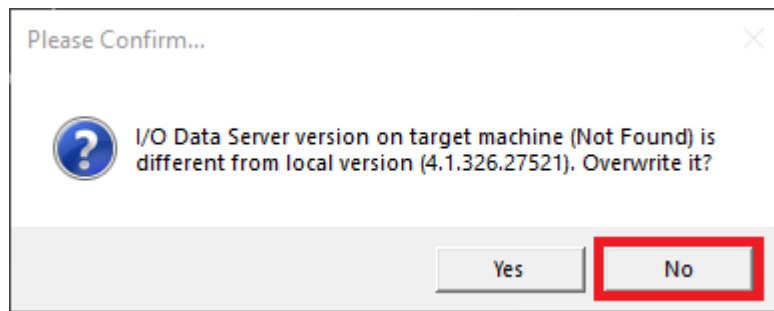
Total upload in progress: 0%

File deploy in progress: 0%

▼ Remote Device Info

Close

6. Click on 'Connect' and confirm to use a secure connection.
7. Refuse deploy "I/O Data Server" and "WebHMI Server" and accept deploy current project to target machine.



8. The project is copied in the target machine's destination folder.



## 4. I/O Data Server

### 4.1. Data Server Introduction

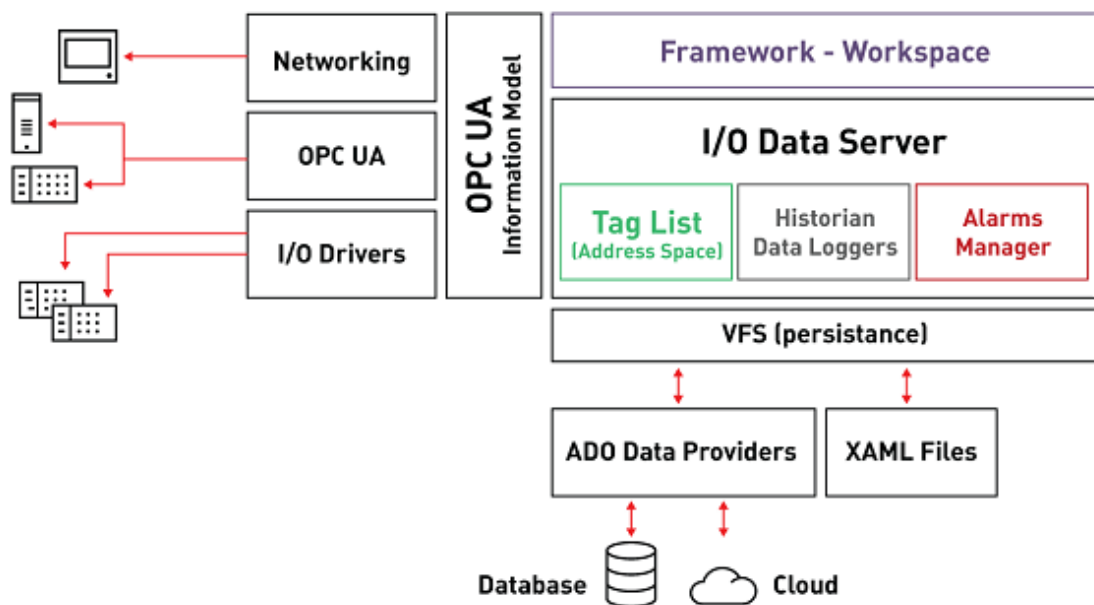
The Movicon.NExT I/O Data Server module is an essential feature to the platform's projects as a communication and data manager server. The Data Server manages communication with the field and third party systems by defining and centralizing information dynamically using Tags described in the project's Tag List resource (Address Space).

In addition the Server manages alarms and records events in the Historical Log.

Ultimately, the Data Server can historically data log process data collected in Tags using the Historian and Data Logger archive manager as an option.

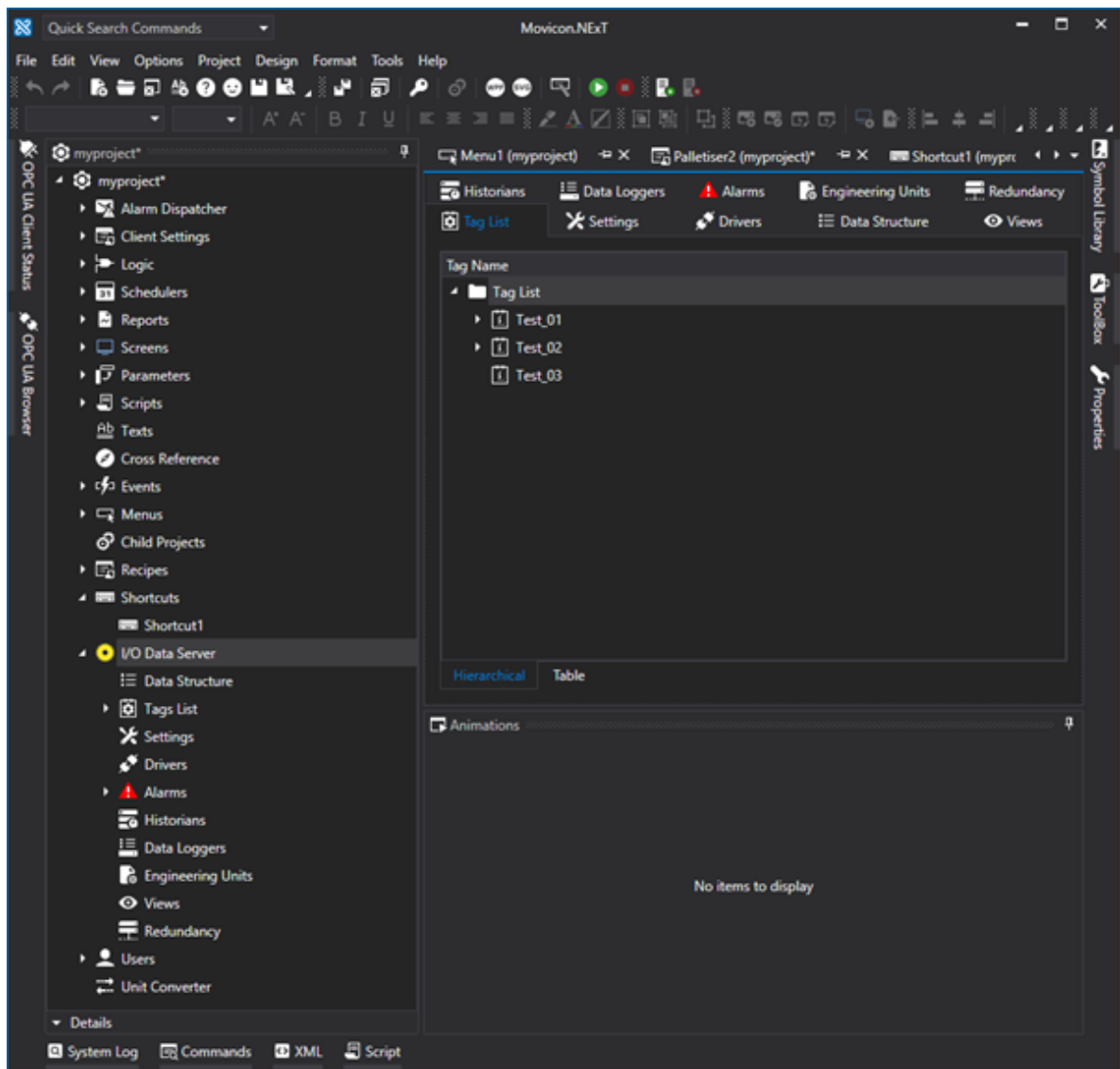
As shown below, the Movicon.NExT Server module is designed to manage:

- **Data Communication**
- **Alarm Manager**
- **Historical Log management**



*This diagram shows the structure of the Data Server which manages communications, the alarm manger and historicals.*

The I/O Data Server module is available from the project resource tree structure. When opening this resource, the workspace open up displaying the general settings and providing access to all the Server's functionality configurations that include the Tag List and others provided with the module as described in this guide.

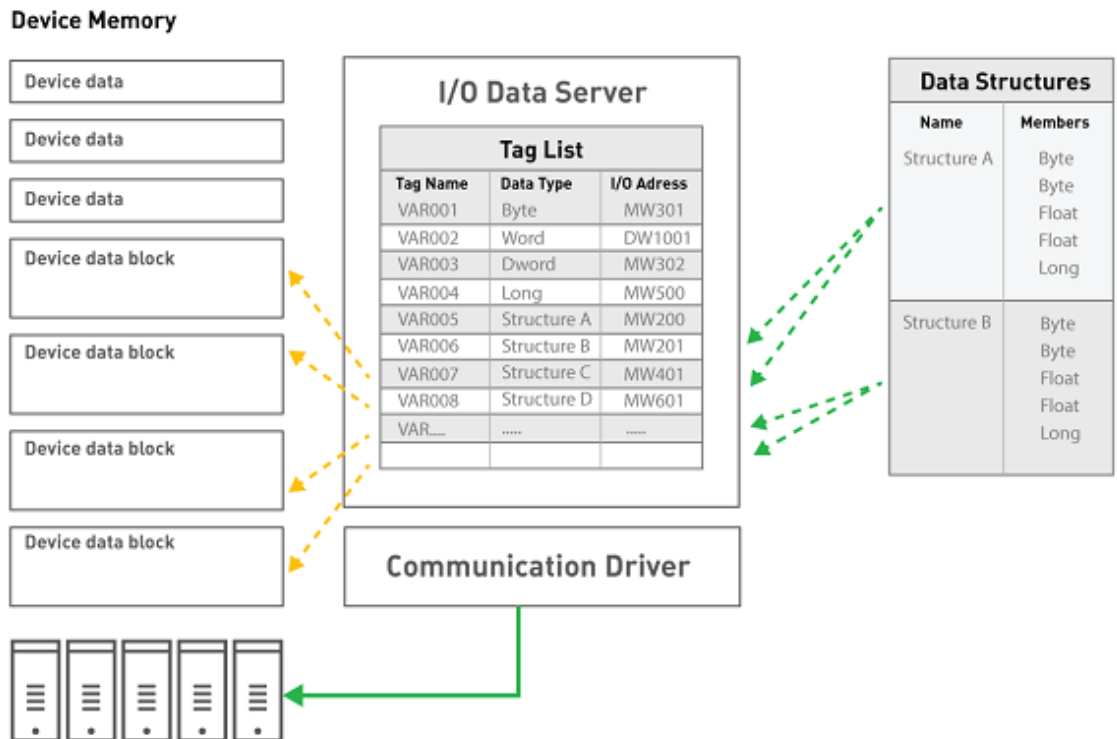


The Tag List can be set as pleased to display in a tree structure containing Tag folders listed in hierarchical order or in a grid to allow operations such as selecting and filtering to be performed.

## 4.2. Data Structures

The I/O Data Server's Data Structure is used to add and configure freely configurable data models defined as "Structure" or "UDT" (User-defined Data Type).

The data structures are used to define complex tag types which have models composed of a series of components called members.



Members and folders can be added within a Data Structure at multiple nesting levels in order to organize the structure members better. The tag's structure defined in this way will then be deployed by the Server to Clients.

Structure members can also be defined structure types which will therefore become Structures of Structures.

However, it is not possible to define members with structures that are recursive.

For example, if you have a Structure called "**Structure A**" which includes a member called "Member1", this "Member1" cannot become a "**Structure A**" Structure type.

Or if you have a Structure called "**Structure A**" which includes a "Member1" member which is "**Structure B**" type, the "**Structure B**" structure cannot have a "**Structure A**" type member.

A member can also be defined as a "**Method**" type. For example, by using the "Demo" Communication Driver and defining a member as the Driver's Method, this method will then be applied only to those variables belonging to that structure and not to all the other Driver variables. In this way, a Structure can be created with a simulation variable and two Driver Start and Stop simulation methods and then afterwards each variable defined with this Structure will be able to manage the stop and start simulation commands independently.

Structure members have a "**MemberOrderID**" property which is used for defining the order with which the members are to be exchanged by the communication driver. The order is established using IDs which are inserted using an incremental number sequence starting with 0 for default. By numbering all the structure members starting from 0, as well as those nested in folders, will enable you to establish the desired sequence with which members will be exchanged with the Communication Driver. There are also two commands with the right mouse click to move the ID of a member more or less.

To enter a new Structure in the project, please proceed as follows:

1. Open the "Data Structure" resource with a double click of the mouse from the Project window, so that the list of project Structures is displayed in the work area. Using the "Add new structure" command, available in the menu Project > I / O Data Server > Data structures, add a Structure to the list.

2. After having executed the command for inserting a new Structure, a window will show for entering the Structure's name. This window can also be accessed afterwards by double clicking on the Structure when inserted.
3. To add new Member variables within the Structure select the Structure and then execute the right click command "Add new Member". At this point a window will appear, which is very much like the one used for inserting variables, for defining the member's properties. Structure member properties are in fact a subset of variables.

### Structure Member Properties

Each Member of a Structure can be configured through the Property Window in the same way as simple variables. The only difference is that a few properties such as the 'I/O Physical Address' and 'Retentive' properties are not available for Structure's members but which can be applied to Structure Variables defined with the Structure.

The Structure can be used to create several Structure Variables once it has been defined and each one of its Members have been configured. It is also possible to intervene to customize the Structure Variable's members to make it different from the other Structure Variables even though defined for the same Structure. A Structure Variable can be exploded to view its structure and access the properties of each individual member. All the Members have a "UseShared" property which is left checked so that they inherit the configuration defined in the Structure. When this property is unchecked, it will be possible to edit some of the Member's properties which will then have priority over the Structure settings. The following modifications can be done when unchecking a Structure Variable's Member's "UseShared" property:

- Insert the I/O Physical Address in each Member
- Insert a Description
- Insert an initial value
- Assign an Engineering Unit
- Assign a Historical
- Define User Access settings: User Level, Access Area in Read and Write, Access mode
- Assign one or more Alarms
- Assign one or more Views
- Assign script code to the Member

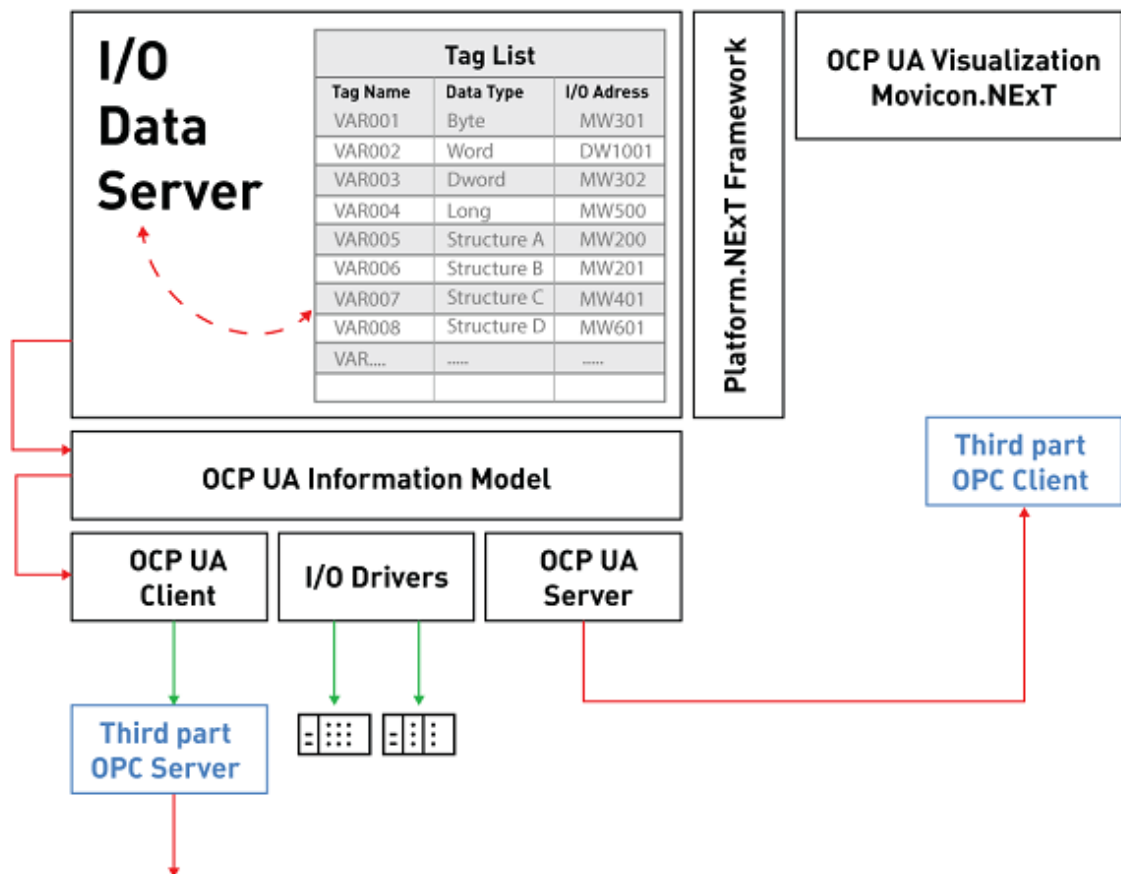


A project that uses the properties of Members defined at Structure Variable level (by disabling the "UseShared" property) will not be compatible if opened with the 3.0 or previous versions of Movicon.NExT.

## 4.3. Tag List

### 4.3.1. Introduction (Tags)

The Movicon.NExT I/O Data Server module is the communication and data management server which is essential for the platform's projects. The Data Server manages communications with the field and with third party systems by defining and centralizing dynamic information through the use of Tags which define variables on the project's Tag List.



*This diagram shows the structure of the system handling data defined using the Tag List while in realtime communication with the field.*

The I/O Data Server module enables the platform to communicate with the field devices in real-time, using the included communication drivers or the connection technology to third party OPC UA servers. All the gathered information is defined in the Server's Address Space, which defines the list of Tags that are visible to all the resources and modules of the entire Movicon.NExT platform.

### 4.3.2. Tag List

Project Variables or Tags are defined in the platform's I/O Data Server module, and in particular are added to the Server's 'Tag List' window.



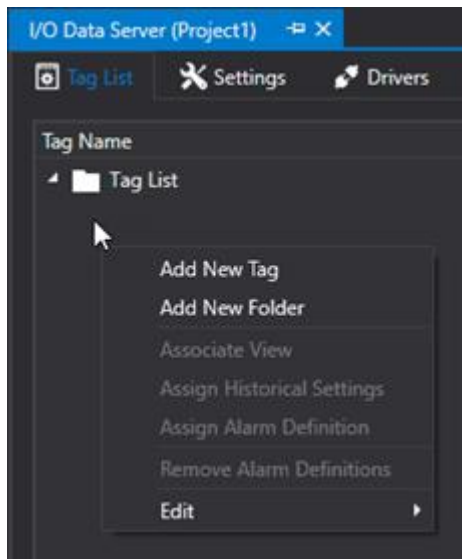
The variables that are not connected to the field (those with undefined "I/O Physical Address" properties) will be considered as Internal Server Variables and will be excluded from the license count.

Variables/Tags can be grouped in folders and subfolder. Any Variable/Tag belonging to different folders can also have the same name. One variable can be defined with different types of data. Variables/Tags can be retentive or can have an initial value.

To add a new variable/Tag in the project, simply open the "Tag List" resource, from the "I/O Data Server" group, within the workspace and use the "Add new Tag" command from the contextual menu, toolbar or the Project > I/O Data Server > Tag menu.

You can also import Variables/Tags from PLC projects by using the importer of the driver being used.

After adding a new variable/tag, a settings window will open with the variable's main properties. This window can also be accessed afterwards by double clicking on the variable inserted.

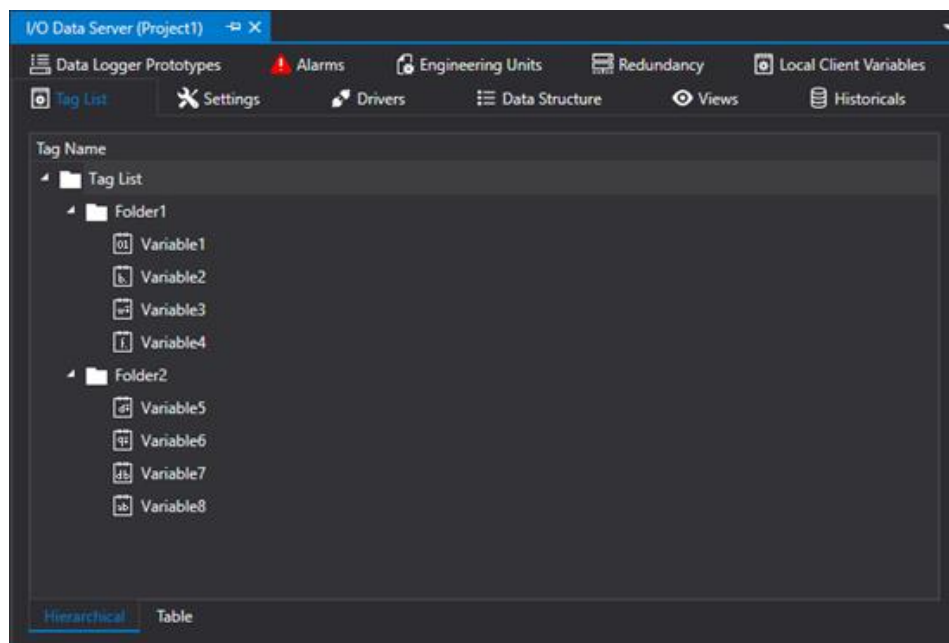


## Displaying the Tag List

The Tag List window can be displayed in two different modes as described below. The selection of one display mode is done by using the selector buttons placed on the bottom border of the same window.

### Hierarchical View (Tree structure)

The variables/tags are displayed in the classic tree structure mode with folders and subfolder groups.

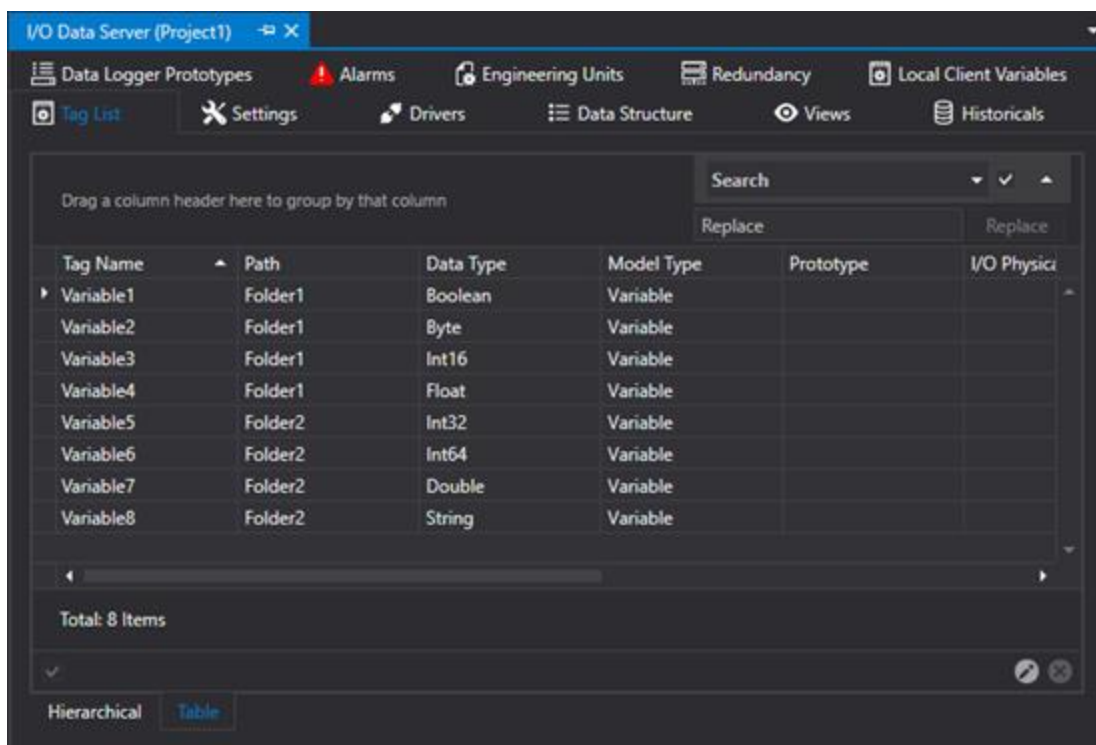


### Table View (Grid)

The variable/tags are displayed by means of using a tabular grid through which properties can be edited and through which all the operative commands that are typical of Data Tables are also available, such as sort bys and filters.

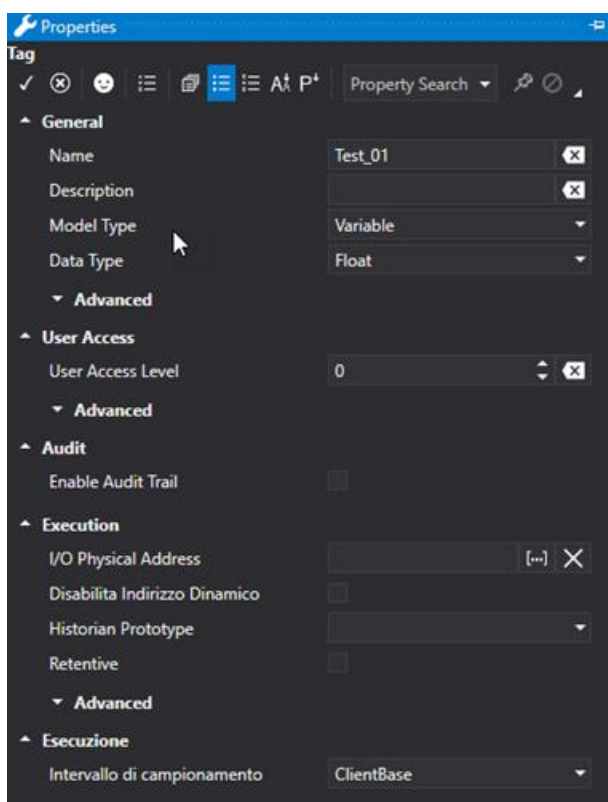


The Search&Replace fields can be found at the top right of the table view, as shown in the below screenshot, and are used for the quick and easy renaming completely or in part of variables.



### 4.3.3. Tag Properties

Each individual Tag inserted on the I/O Data Server's Tag list can be configured by using its **Property Window**. This window opens when double clicking the tag selected.



## General Properties

The General properties allow you to configure the Tag's fundamental properties.

### Name

The name of the variable is inserted in this field. The variable must have its own unique name when in a group.



Names of tags can only contain alphanumeric characters and the underscore character '\_', however the underscore and numbers cannot be used as the initial character.

### Description

This field is used for giving the variable a description (field not obligatory).

### Model Type

This field is used for selecting which type of variable model is being inserted, whether an analogic value, method or an enumerator, etc.

#### Variable

Tags with "variable" model type represent tags that can be associated to common "Data Types" without the possibility to associate an Engineering Unit.

A "Variable" tag type does not support the Engineering Unit properties: in this case, the "Analog" Tag type should be used.

#### Variable with Engineering Unit

These are variables that can be associated with an engineering unit.

#### Structure

The "Structure" Tag types are Structure or User-defined Data Type (UDT) variables. When a Tag is defined as a "Structure" type, you will need to select which structure it is to refer to (for further information please refer to the paragraph on "Data Structures").

#### Enum (two states), Enum (multiple states)

The "Enum (two states)" and "Enum (multiple states)" Tag types are similar: both are Enumerative type but the first only has two states while the other has more. Enumeration strings must be specified for both these models.

The "|" character is used in the string to separate text of one state and the next.

#### Method

The "Method" Tag model type allows you to define methods or function which are made available by the project's OPC UA Server.

A "Method" object is not a variable that contains a value but becomes an actual function that can be invoked with appropriate "Call Method" command from the Command List.

### Engineering Unit (Advanced)

This property is only available when "Analog" has been selected as "Model Type".

This field is used for selecting an Engineering Unit which was previously inserted in the 'relevant I/O Data Server window in order to associate it to the variable.



The Engineering Unit will be applied to the Tag value calculating the resulting scale value from the parameters set with the Min. and Max raw value and the Min. and Max. value of the set engineering conversion rate.

### Data Type

this field is used for selecting the variable's data type. This field is only available when selecting specific 'Model Types' such as the "Analog" Model Type. In this case the variable can be entered with data type such as Boolean, Byte, Int32 or Float, etc.

### Use Shared

This property will show when selecting a member of a Structure Type Tag. This property is active for default in members of Structure type Tags.

The Use Shared option indicates that the selected Tag being a member of a Structure, will inherit the configuration directly from the Structure.

For further information please refer to "Data Structures" .

### Enable Tag Statistics

This property is used to enable Tag Statistics. When enabled the following stats values will be published by the Server:

- Min: minimum value obtained by the variable
- Max: maximum value obtained by the variable
- Average: average value obtained by the variable
- NumUpdate: number of variable value variations.
- TotalTimeOn: total time variable maintained a value other than zero.

Variable value statistics are saved in the variable's retentive file, therefore if you enable the Tag Statistics, you will also need to enable the Retentive option.

The Variable Statistics are only supported for scalar tags and not Structure or Array variable types.

The Variable Statistics are also managed in redundant systems.

The EditDisplay object can be used to access Tag value statistics by displaying the Tag's current value or its value statistics.

### Array Dimension

To create a Array tag simply select the number of items it is composed of in the "Array Dimension" field. When left set with the zero value, the tag will remain a simply tag type. Array elements start with "0".

Structure arrays cannot be defined but contrary to this array structures can.



To insert an initial value in a array tag, you must insert the values of each element separated with a space and " |" pipe character, and all enclosed within brackets in this way for example: {1 |10 |5.5 |6 |35} .

### Enum Strings

**This property is only available when the "Enum (two states)" or "Enum (multiple states)" has been selected as the "Model Type".**

The texts corresponding to the enumerator or boolean values are entered in this field. The "|" character must be used to separate texts of one state and the next. The string can be composed using the edit window which can be accessed from this property. This window also allows you to associate strings from the project String Table with the 'Texts' Resource.

## Data Structure

**This property is only available when "Structure" has been selected as the "Model Type".**

This field is used for selecting the structure to be used for defining the variable with. This field is only available when the variable's "Model Type" has been selected as "Object Type". The Data Structure to be selected must be previously defined in the appropriated "Structure" resource.

## Execution Properties

The Tag Execution properties are important because they are used to configure the tag's role within the project.

## Physical I/O Address

Each Tag variable can be associated to a Physical I/O Address through a Communication Driver. In order to do this, you will need to add and configure at least one communication driver.

By using the properties of the Tags' "Physical I/O Address" in the properties window, you will be able to access the address settings through a window whose properties depend on the Driver. Please consult the chapter on "Tag - Physical I/O Address" of each driver for further information.

## Gateway Functionality

A Tag can also be assigned two or more addresses to link it to two or more communication drivers simultaneously.



The Physical I/O addresses to associated to the same variable must be of different Drivers: it is not possible to insert two Physical I/O Addresses belonging to the same one driver.

When defining two or more Physical I/O Address in the same Tag, you will get a Gateway functionality where the change of a value associated to an address read through the communication driver will be propagated to the address associated to another driver.

There are no restrictions as to the connection types that can be used. Therefore, each Physical I/O Address can be Read Only, Read/Write or Write Only.

However, in cases where we have two addresses both in Read/Write, you should keep into account that data may be updated simultaneously in different drivers. In this case, the last read performed will be the one that valorizes the Tag. It is not possible in this situation to assign priority to one or the other driver.



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 will, in fact, overwrite the already existing variable and therefore the initial dynamic link will be replaced with the new one.

## Disable dynamic address

This option for default permits you to disable any Physical I/O Adresses assigned to individual tags, structures or structure members.

When enabling this property, you will not be able to edit the 'Physical I/O Address' field, which will be disabled, and the tag will be ignored while the dinjob file is being created by communication drivers. **The tag will not be counted in terms of runtime license.**

## Historian Settings

This property is used for selecting a Historian type to associate to the tag for recording data. Historians must be previously entered and configured in order to do this. Once a historian has been selected, the tag will be recorded in the historical data archives accordingly.

For further information please see "Historicals".

### Retentive

This check-box is used for defining the variable as retentive, meaning that its last value must be saved on file and maintained for subsequent project startups.



Note that the system memorizes the tag value of each change on disk. We therefore advise you to avoid using retentive indiscrimination as not to compromise the system performances and access to the disk.

### Always in Use (Advanced)

This is used to set the Tag as "Always in Use" and therefore will always be exchanged with the field even when set with a physical I/O adress and not used by the project.

### Enable Redundancy (Advanced)

When disabled, this option excludes the Tag from the redundancy management if provided by the project. In this way the Tag will no longer be synchronized with the other Servers on the redundancy list.



In cases where the variable is Structure type, it is possible to enable or disable the synchronization of individual Structure members (The IsRedundancyEnables property is not present at Structure Variable level but only for the individual members).

### Initial Value (Advanced)

It is possible to assign an "initial" value that the variable will assume at startup, until its value is determined otherwise by drivers or logic.

When the Tag concerned is an array type, the values of each element will be inserted and separated with a space and " | " pipe character, and all enclosed within brackets in this way for example: {1 |10 |5.5 |6 |35}.

### Use Script in separate thread (Advanced)

Enabling this option will run the Tag' script in a separate thread instead of the one normally used.

### Update Time

This is used to set an update time that will be used for sampling the tag. The default selection uses the update value defined in the Client's settings. Different times can be set and whose value is set in the i/O Data Server's settings:

- Client Base: the sampling interval will be the one defined in the "Active Item sampling time" parameter of the Client settings or the one defined in the "Default sampling interval" parameter of the I/O Data Server. In cases where the Tag is in use by the Client and I/O Data Server, the lowest value of the two previous parameters will be used.
- Slow/Medium/Fast: The sampling interval will follow the selected family. The value of each interval sampling family can be configured in the I/O Data Server's advanced settings



The sampling interval may not be respected in cases where the Tag is connected to a driver which used an absolute address (such as for example the TCP/IP Modbus). In these cases, the aggregation of the read requests by the driver, might influence the sampling interval.

## User Access

The Tag's 'User Access' property allows you to configure its security properties.



A tag's access rights are applied the moment the tag is subscribed to the Server. This means, for example, to enable the opening a Screen that connects to a specific Tag, the user must have already logged in with the appropriate credentials needed to access that Tag. Otherwise the Tag will be subscribed but not accessible. At that point it will not be sufficient to log in again with the appropriate credentials because the Tag will have already been subscribed to the Server. Therefore you will need to close the screen and wait for the time needed to remove the Tag from the Server and then log in correctly in order to re-open the Screen and subscribe the Tag again.

The management to read/write access Tags associated to objects can be assigned to the Server project. To activate this functionality you will need to enable the "User Management" -> Access Level from Tag' property in the desired objects.

#### User Read Access Mask

This is used to select up to 32 access areas from a selection box to give or deny the logged in user access to read data. The user in question must however have a 'Access Level' that is equal to or higher than the one requested by the Tag.

#### User Write Access Mask

This is used to select up to 32 access areas from a selection box to give or deny the logged in user access to write data. The user in question must however have a 'Access Level' that is equal to or higher than the one requested by the Tag.

#### User Access Level

This is used to define a number corresponding to the hierarchical access level, to allow those users who Log On with an access value equal to or higher than this number to have access to and operativity with the Tag value. The access modes available to the authorized user are those defined by the "Access Mode" property.

Once the first User authentication phase has concluded successfully by logging in with the appropriate Access Level, the user can still be prohibited from performing certain operations on the Tag by using the Access Area in read and write.

#### Access Mode

This setting is used to define the access mode to use by the Server to publish the Tags. When authorizing a user access to the Tag, it is also possible to establish which operations they will be allowed to do using this property.

The possibilities are:

- None: User will not be allowed to write or read data independently from their access rights used to connect to the Tag.
- Read: User will not be allowed to write but only read data independently from their access rights used to connect to the Tag.
- Write: User will not be allowed to read ut only write data independently from their access rights used to connect to the Tag.
- Read/Write: User will be allowed to read and write data according to their access rights used to connect to the Tag.

## Audit

#### Enable Audit Trace

This enables the Audit function. This option is also available for structure members but not for methods. The Audit traces each tag change made by the user while operating with the supervisor: display, check box and selectors. The trace and relative comment (if Enter Comment On Audit) has been enabled), will be recorded in the table already in use by the Historian. The comment in the historian will be saved in the new 'Reason' field.

#### Enter Comment On Audit

Used to enable the comment on the variables' Audit. When activating this property, the 'Force Password On Audit' and 'Access Level Required to Confirm' properties will automatically show.

#### Force Password On Audit

Even though the user has already logged-on, this forces the user to re-enter their password again to confirm the comment they have entered in the appropriate audit window which appears on the client.

#### Access Level Required to Confirm

Minimum level required for double checking Audit. Therefore, if the user who entered the comment has lower level access, a double check will be made by requiring another user with this level to confirm.

#### Max Audit Age

The maximum age for Audit Trace information storage. Rows will automatically be deleted once exceeding this age.

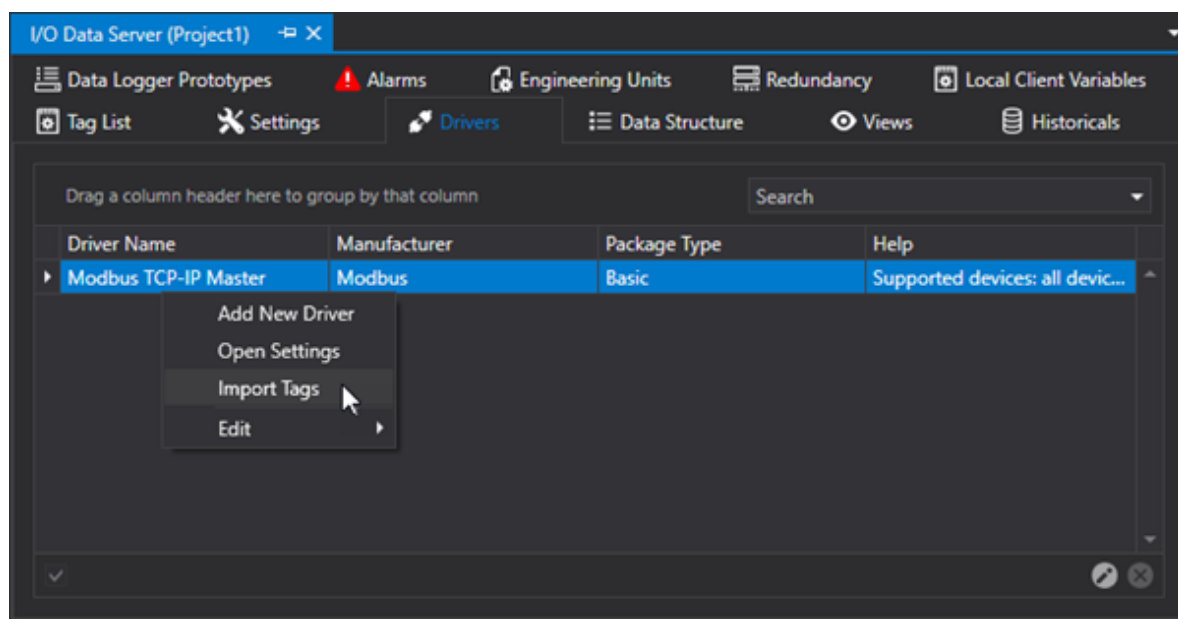
### 4.3.4. Import/Export

The development environment consents to the Importing and Exporting of Tags according to the different modes described below.

#### Import from PLC

Most of the Communication Drivers provide a tool to import tags from the PLC project, or from an export performed by the PLC programming software.

To import tags in this way simply open the I/O Data Server's Drivers resource and select the driver from which you wish to import the Tags. Then select "Import Tags" from the contextual menu as shown below.

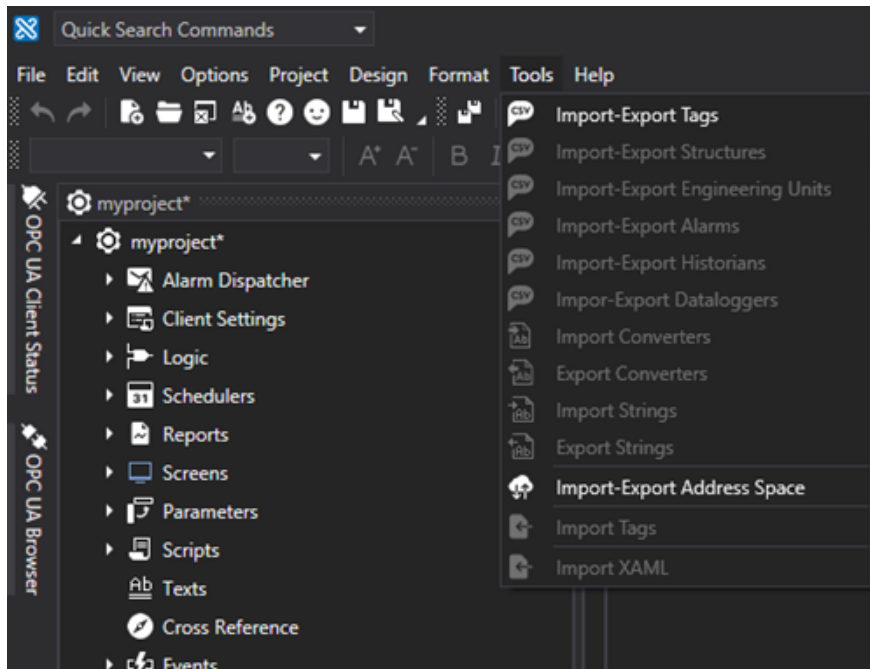


#### Tag List Import/Export

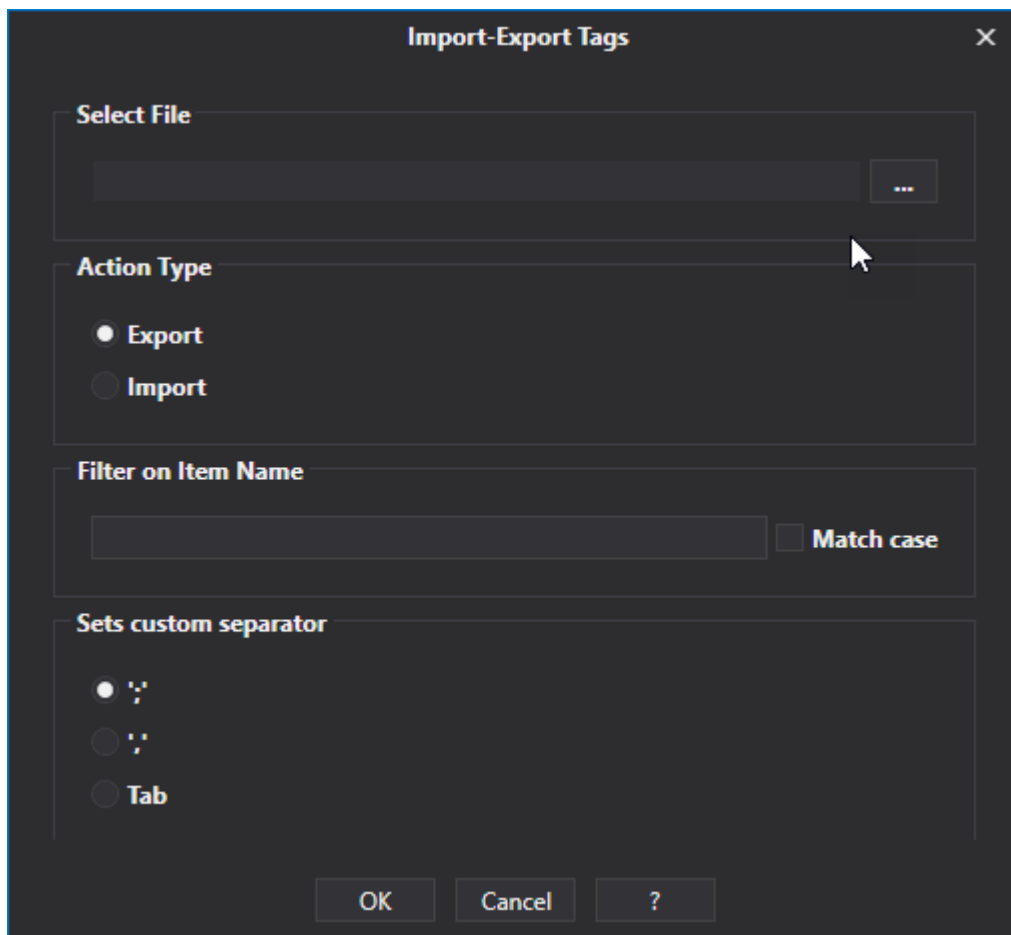
The I/O Data Server allows you to import/export all the Tag List contents from/to a .CSV file. This will allow you to manage/edit Tag with an external editor such as Microsoft Excel.



The import/export of the Tag List also includes any Structures, Engineering Units and Alarms associated to the Tags.  
The Import/Export window is reachable from the Tools > Import-Export Tags menu.



This results as shown in the screen shot below:



Select File

Use this to setting to select the file to use for importing or exporting Tags.

Action Type

This is used to select whether to Export or Import tags.

Filter on Item Name

Use this to filter data to import or export. It is also possible to define whether the command is to be upper or lower case character sensitive.

Set custom separator

This is used to select the separator type to use between one tag and another.

4.3.5. Server System Tags

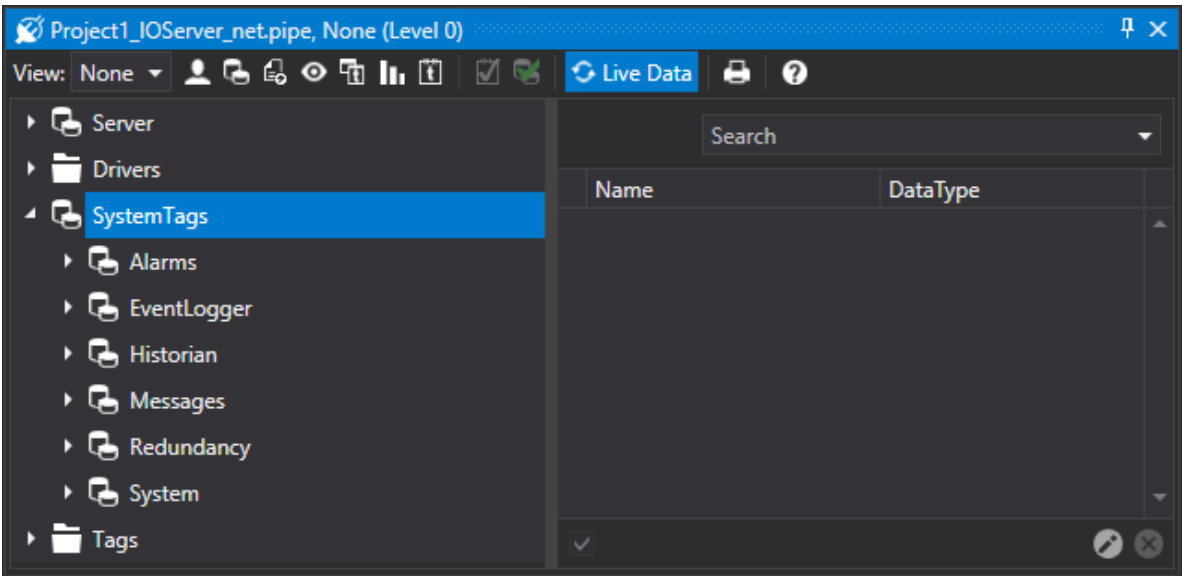
The Movicon.NEXt Data Server defines a list of tags which are available for entire project use on both client and server. However, there are Tags that are only available for use on the Client side. These tags are very handy to use in certain situations when the Data Server is not used. The Tags that can be used only when selected on the Client side are:

- **System Tags**
- **Local Tags**

Both types are described in the chapter on Client Visualization.

System Tags

The Server's System Tags relating to the project are exposed in the "SystemTags" folder as shown in the screen shot below:



The below table list all the exposed variables with their descriptions:

Group	Tag Name	Data Type	Description
Alarms	SoundState	Boolean	Tag for reading or setting the alarm sound state to activated or deactivated.
	NumActiveOff	Int64	Number of alarm active with OFF status.
	NumActiveOn	Int64	Number of alarm active with ON status.
	NumActiveOnOff	Int64	Number of alarm active with ON and OFF status.
	NumEnabled	Int64	Number of enabled alarms.
	NumNotAck	Int64	Number of acknowledged alarms.
	NumShelved	Int64	Number of shelved alarms.
Eventlogger	DeleteEntriesPending	Integer	Number of pending delete entries to be performed.
	DeleteEntriesRunning	Integer	Number of Delete entries being run.
	DischargingEntries	Boolean	"True" when discharging data entries due to reaching internal memory cache limit (Max Pending Entries).
	FailsEntriesPending	Integer	Number of failed entries pending database recording.
	FailsEntriesRunning	Integer	Number of previously failed entries being recorded on database.
	FlushEntriesPending	Integer	Number of entries pending flush to local XML file after the maximum number of attempts failed to record on Database.
	FlushEntriesRunning	Integer	Number of entries being flushed to the local XML file to then be retrieved by the system when

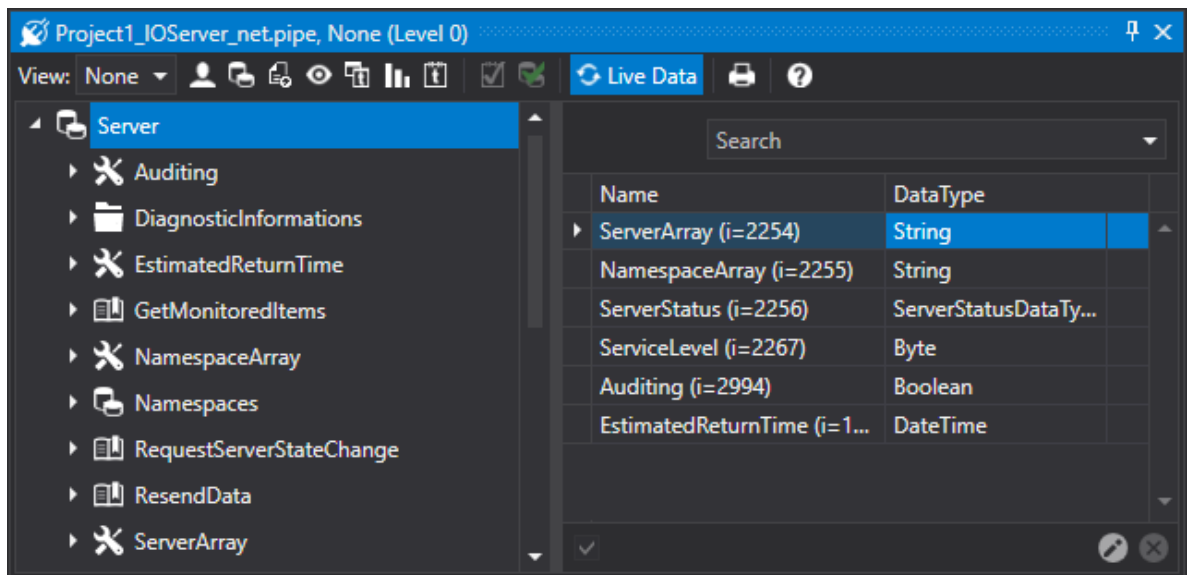


			database re-entered from error state.
	RecordEntriesPending	Integer	Number of entries pending database recording.
	RecordEntriesRunning	Integer	Number of entries being recorded on database.
Historian	DeleteEntriesPending	Integer	Number of delete operations pending execution.
	DeleteEntriesRunning	Integer	Number of delete operations being run.
	DischargingEntries	Boolean	"True" when discarding data entries due to reaching the internal memory cache limit (Max Pending Entries).
	FailsEntriesPending	Integer	Number of entries waiting to be recorded on database and which failed the previous attempt to do so.
	FailsEntriesRunning	Integer	Number of entries being recorded on database and which failed the previous attempt to do so.
	FlushEntriesPending	Integer	Number of entries waiting to be flushed to the local XML file due to exceeding the maximum number of attempts after the last attempt to be recorded on database failed.
	FlushEntriesRunning	Integer	Number of entries being flushed to the local XML file to then be retrieved by the system when database returns back from error state.
	RecordEntriesPending	Integer	Number of entries waiting to be recorded on database.
	RecordEntriesRunning	Integer	Number entries being recorded on database.
System	DynamicTags	Integer	Returns the number of TAGs counted for license purposes. Each 'simple' tag from 1 to 64 bit is counted as a licensed TAG while

			each member of an array or structure tag counts as 1 TAG.
Redundancy	RedundancyActiveServerState	Boolean	Indicates whether the local server is active or inactive. Each server indicates its on status.
	RedundancyActiveServerHostName	String	This tag is redundant in all the Servers that have been started up and reports the name of the Active Server which is the one in control.
	RedundancyArrayAliveServerHostName	String	This tag is redundant in all Alive Servers and reports the list of names of those Servers which are currently alive.
	RedundancySwitchActiveServer	Method	This method, which can be executed both in the Active Server and inactive Server, switches the Active Server when invoked. This method is enabled when there are two Servers on the list only and if the 'Keep Server Active' redundancy property has been enabled.
	RedundancyActiveServerHostName	String	Hostname del server di ridondanza attivo.
	RedundancyArrayAliveServerHostName	String	Lista degli hostname dei server di ridondanza disponibili.
	RedundancySwitchActiveServer	Method	Metodo che permette di cambiare il server di ridondanza attivo.

## OPC-UA Server Tags

Server Syster Tags exposed according to the OPC UA specifications can be found in the "Server" folder as indicated below:



The below table lists the exposed variables with their descriptions:

Group	Variable Name	Data Type	Description
Server	ServerArray	String	Defines an URI server Array. This variable is also defined in the Server table. Each URI in this array represents a unique global logic name for a Server in the network environment in which it is installed. Each OPC Server instance has an individual URI used in the table of servers of other OPC UA Servers. The 0 index is reserved for the URI of local ServerObjects. Values greater than 0 are used to identify remote servers and are specific to a particular server.

	NamespaceArray	String	<p>Defines an Array of the URI "NameSpace". This variable is also defined as "NameSpace" Table. The indexes in this table are called <b>NamespacelIndexes</b>. The "NameSpaces" are used in Nodelds of OPC UA services, instead of in the URI of the space of longer names. The 0 index is reserved for the OPC UA "NameSpace", The index 1 is reserved for the local server. Clients can read the whole table or individual items in the table. The server must not modify or remove any of the items from this table while clients have a session opened in the server, since they memorize this table in their cache. However, a server can add items to the table even when clients are connected to the server.</p> <p><b>Therefore it is advised that servers do not modify the table's indexes but only add items, because clients can memorize the Nodelds in the cache using the indexes.</b></p> <p>The Clients that memorize the NamespacelIndexes of NodeldsI in the cache should always check that the NamespacelIndexes are not already present in the cache when starting up a session.</p>
	ServerStatus	String	<p>This contains elements that describe the state of the server.</p>

	ServiceLevel	Byte	This describes the server's capacity to provide data to clients. The value range is from 0 to 255, where 0 is the worst and 255 is the best. This is intended to give clients an indication of the redundant servers' availability.
	Auditing	Boolean	This is a boolean value that specifies whether the server is currently generating Audit events. It is set to TRUE if the server generates control events otherwise it will be set to FALSE.
	EstimatedReturnTime	DateTime	Indicates the estimated time with which the server will return back to being available on the list of active servers.
Server Capabilities	ServerProfileArray	String	Defines the server's conformity profile.
	LocaleIdArray	LocaleId	This is a Local Array supported by the Server. The server may not be aware of all the supported LocaleId if systems or underlying devices, that have been granted access, do not report the LocaleId supported.
	MinsupportedSampleRate	Duration	This defines the minimum sample rate, including 0, that is supported by the server.

	MaxBrowseContinuationPoints	Duration	<p>This is an integer that specifies the maximum number of "Continuation Points", parallel with the Browser service, that the server can support per session. The value specifies the maximum continuation points that the server can support under normal circumstances, therefore there is no guarantee that the server will be able to support this maximum. The client must not open multiple Browse calls with more "Continuation Points" open in respect to those exposed by this variable. The 0 value indicated that the server does not place any limit on the number "Continuation Points" in parallel to those that can be used by the client.</p>
	MaxQueryContinuationPoints	UInt16	<p>This is an Integer that specifies the maximum number of "Continuation Points" in parallel with QueryFirst services that the server can support per session.</p> <p>The value specifies the maximum continuation points that the server can support under normal circumstances, therefore there is no guarantee that the server will be able to support this maximum. The client must not open multiple QueryFirst calls with more "Continuation Points" open in respect to those exposed by this variable. The 0 value indicated that the server does not place any limit on the number "Continuation Points" in parallel to those that can be used by the client.</p>

	MaxHistoryContinuationPoints	UInt16	<p>This is an Integer that specifies the maximum number of "Continuation Points" in parallel with ReadHistory services that the server can support per session.</p> <p>The value specifies the maximum continuation points that the server can support under normal circumstances, therefore there is no guarantee that the server will be able to support this maximum. The client must not open multiple ReadHistory calls with more "Continuation Points" open in respect to those exposed by this variable. The 0 value indicated that the server does not place any limit on the number "Continuation Points" in parallel to those that should used by the client.</p>
	SoftwareCertificates	SignedSoftwareCertificates	<p>This is a software certificates Array containing all the certificates supported by the server. this must be the same list certificated returned by the "CreateSession" service. The certificates in this property are not signed.</p>
	MaxArrayLength	UInt32	<p>The maximum Array value length supported by the server.</p>
	MaxStringLength	UInt32	<p>The maximum string value length supported by the server.</p>
	MaxByteStringLength	UInt32	<p>The maximum Byte string value length supported by the server.</p>

Server Configuration	ServerCapabilities	String	Defines the capabilities supported by the OPC UA server.
	SupportedPrivateKeyFormats	String	This specifies the PrivateKey formats supported by the server. The possible values include "PEM" (see RFC 5958) or "PFX" (see PKCS # 12). The array is empty if the server does not consent external clients to update the PrivateKey.
	MaxTrustListSize	UInt32	This is the maximum size of the Trust List in bytes. 0 signifies no limit. The predefined setting is 65 535 bytes.
	MulticastDnsEnabled	Boolean	When TRUE, the server will be announced using a DNS multicast (this can be modified by writing in the variable).
Server Diagnostics	ServerDiagnosticsSummary	ServerDiagnosticsSummaryDataType	Contains Diagnostics summary for the server.
	SubscriptionDiagnosticsArray	ServerDiagnosticsSummaryDataType	This is a diagnostics subscription information Array. There is an item for each notification channel effectively established in the server. Its TypeDefinitionNode is the VariableType SubscriptionDiagnosticsArrayType, which provides a variable for each array item. These variables are also used as variables to which other variables can refer to.



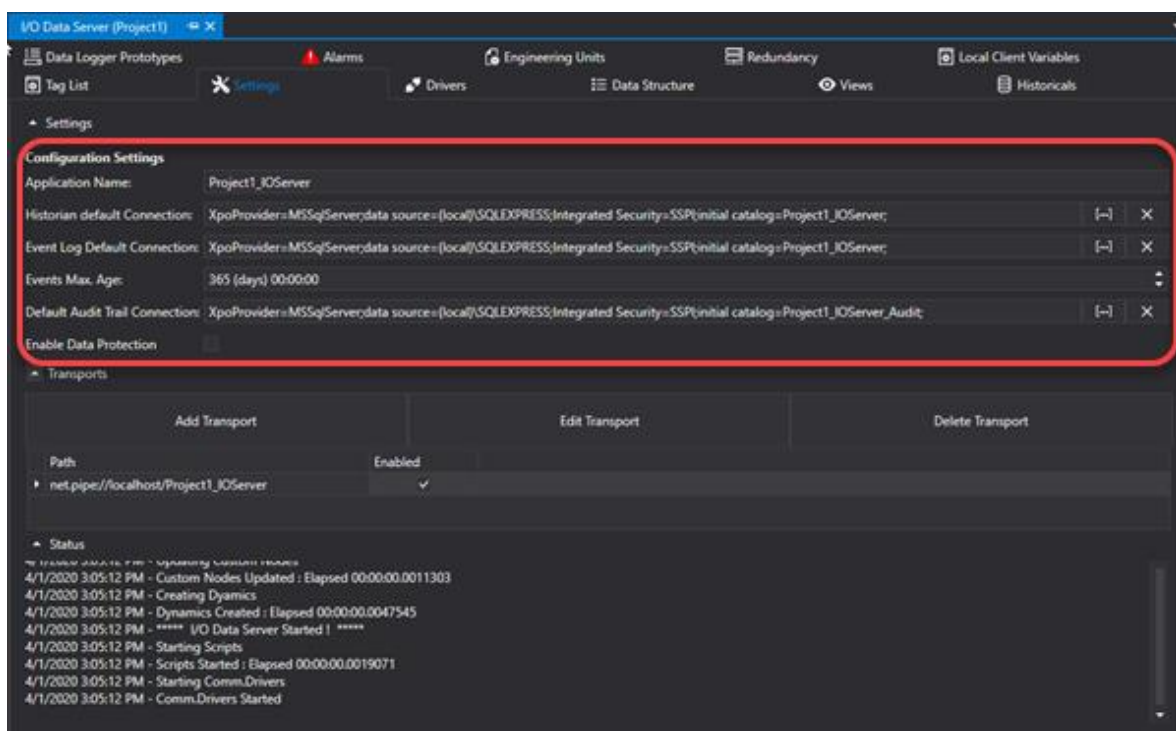
	EnabledFlag	Boolean	<p>This identifies whether or not the server collects the diagnostics information. It can also be used by clients to enable or disable the server's diagnostics information collection.</p> <p>The following boolean value settings are applied:</p> <ul style="list-style-type: none"> <li>• TRUE indicates that the server collects diagnostics information and when value is set to TRUE, the collection of diagnostics information is restored and enabled.</li> <li>• FALSE indicates that no statistical information has been collected and setting the value to FALSE disables the collection without restoring the statistics values.</li> </ul>
Server Redundancy	RedundancySupport	RedundancySupport	This indicates which redundancy is supported by the server.
	CurrentServerid	String	This is a value that is needed to identify the currently used server in the redundancy set.
	RedundantServerArray	RedundantServerArrayDataType	This is a value that contains an array of servers available in the redundancy set.

	ServerUriArray	String	This is an array with the URIs of all the OPC UA Server-s redundant servers. As this is a nontransparent redundancy ambient, the client responsible for the subscription to redundant servers, may or may not be able to open a session with one or more redundant servers of this array.
	ServerNetworkGroups	NetworkGroupDataType	This contains an Array of NetworkGroupDataTypes. The URIs of servers of this array (in the structure's serverURIs) must be exactly the same as those provided in the ServerUri Array despite in a different order. Therefore, the array should represent a list of HotAndMirrored redundant servers. If one server only supports the network redundancy, it will have only one item in the ServerNetworkGroups. The NetworkPaths in the structure present the redundant network paths for each server. The NetworkPaths describe the different paths (one item for each path) ordered by priority: Each network path contains an endpointUrlList with a string matrix, each one containing an endpoint URL. This will allow different protocol options to be used for the same network path.

## 4.4. Settings

### 4.4.1. General Server Settings

The Server's **General Settings** window is used to configure the general settings that determine how the Data Server functions.



## Configuration Settings

The I/O Data Server's Configuration Settings Window contains the module's general settings. A default value has been preset for each individual setting parameter, which for most cases, is consider optimal. However, the parameter can be changed according to the user's specific needs.

### Application Name:

The Name of the Server application. This name can be customized and will be displayed in the selection list when browsed by Clients (eg. third party OPC UA Clients).

In addition to the Application Name other information such as "Manufacturer Name", "Product Name", "Product URI", "Software Version" and "Build Number" can also be entered. This information will then be published by the Server as additional information or for managing Security Certificates.

Please refer to information on "Renamed App Name" in this page

### Default Historian Connection

This Connection String to the database is used to record data from the Historians, Dataloggers and Recipes. It is also possible to specify a different connection string to replace the one for default within the configuration of each Historian, Datalogger or Recipe. The connection string's configuration can be done by using the configuration guide below:

Or you can define the connection string manually as described in the "DB Connections Strings" topic.



As an alternative, you can define the connection string manually as described in the topic on "Database Connections"

### Default Log Event Connections

Connection String to the database that will be used for recording Events such as: Alarm Log, System Events, Driver Events.

The configuration of connection strings can be done using the below configuration guide:

Or the string can be defined manually as described in the "DB Connections Strings" topic.



As an alternative, you can define the connection string manually as described in the topic on "Database Connections"

### Max. Log Event Period

The maximum duration of historical data recorded in the Database of Events and Alarms (Historical Log) permits you to adjust the period of time that data should be kept in the archives. The default is set at 365 days and means that once this period of time has been reached, the system will begin recycling therefore overwrite the oldest data.

### Default Audit Trail Connection

Connection String towards the database that will be used to record Tag Audit Trail information.

For further information, please see the chapter on CFR21 Part 11

The connection string configuration can be done by using the configuration guide below:

The screenshot shows a configuration window for a database connection. It has four tabs: 'SQL Server' (selected), 'SQL Azure', 'MySQL', and 'SQLite'. Under the 'SQL Server' tab, there are four input fields: 'SQL Server:' with a dropdown menu showing '(local)\SQLEXPRESS', 'Database:' with a dropdown menu showing 'Project1\_IOServer', 'Login name:', and 'Password:'. Below these fields are two radio buttons: 'Windows Authentication' (which is selected) and 'SQL Server Authentication'. At the bottom, there is a text box labeled 'Connection String' containing the following text: 'XpoProvider=MSSqlServer;data source=(local)\SQLEXPRESS;user id=;password=;initial catalog=Project1\_IOServer;Persist Security Info=true'. At the very bottom of the window are three buttons: 'OK', 'Cancel', and a help button with a question mark.

Or the connection string can be defined manually as indicated in the "DB Connections Strings" topic.



As an alternative, you can define the connection string manually as described in the topic on "Database Connections"

### Enabling Diagnostics

This is used to enable the publication and management of diagnostics information on the server to which clients can access.



Attention! The activation of this option might have an impact on the server's performances. Therefore please use with caution and only when strictly necessary for debugging purposes only. For example, when enabling this option in projects containing vast amounts of Tags defined in the Server's Address Space, the Server may take time to startup.

### Enable Data Protection

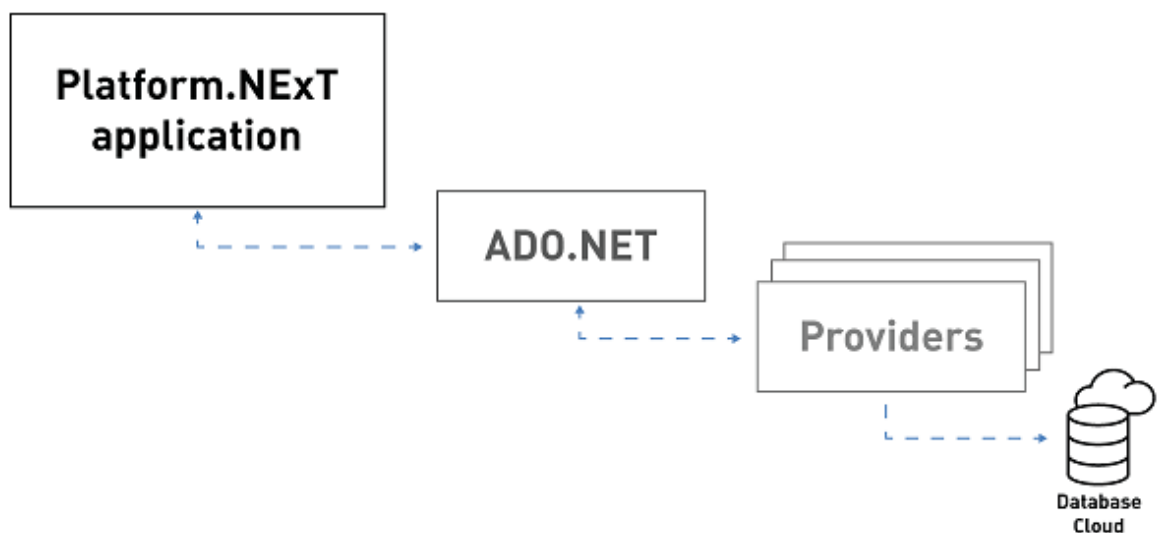
When enabling this function, Movicon.NExT will use the system User ID (encrypted) and Transaction Log combination to ensure integrity of each data recorded on database (record). Thanks to this control, Movicon.NExT will be able to ensure users with original recorded data impeding any possibility of data manipulation or deletion.

For further information please consult the chapter on CFR21 Part 11.

### 4.4.2. Database Connections

The Movicon.NExT I/O Data Server module connects to databases using ADO.NET by means of two Data Providers: the Xpo Provider or the .NET Framework Provider.

Thanks to these two Data Providers, the application is completely independent from the type of database used, and the connection strings can be customized in the appropriate configuration properties in order to use your preferred database.

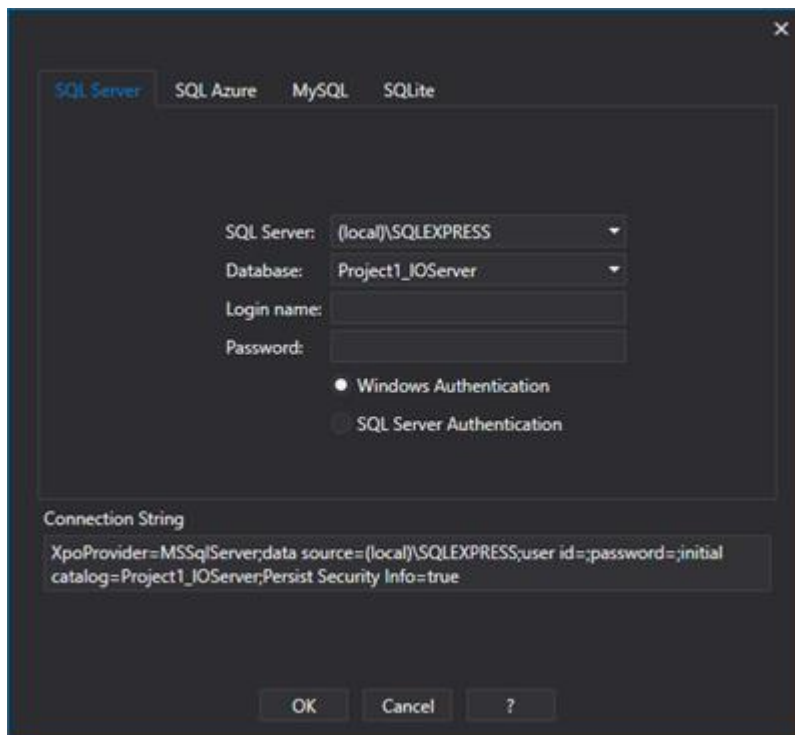


The use of one Data Provider compared to another one depends on which resource accesses the database:

- Xpo Provider = Historian, Event Log, Audit, Recipes
- NET Framework Provider = Datalogger

### Xpo Provider Connection Strings (Historian, EventLog, Recipes, Audit)

The Xpo Provider connection strings are used by the Historian, Event Log, Audit and Recipes resources. They can be defined using the wizard shown in the screen shot below:



This wizard allows you to quickly configure the indicated default databases:

- SQL Server
- SQL Azure
- MySQL
- SQLite

As an alternative, you can also use other DBs whereby the Xpo Provider connection strings must be defined manually and then be inserted in the "Default Historical Connection", "Event Log Connection", "Default Audit Trail Connection" of the Advanced Server Settings.



Movicon.NExT fully supports connections with the below listed DBs. For connections with the other DBs listed in the table below, please visit the DevExpress website:

- SQL Server (from version 2012 to version 2019)
- SQL Server Express (from version 2012 to version 2019)
- MySQL (version 8.0.26)
- MYSQL Community Server (version 8.0.26)
- MariaDB (from version 10.3.31 to version 10.4.21)
- SQLite (from version 3.30.0 to version 3.36.0)

Check to see that the installed DevExpress version in the "C:\Program Files\Progea\Movicon.NExT 4.1" path matches the version displayed on the DevExpress website.

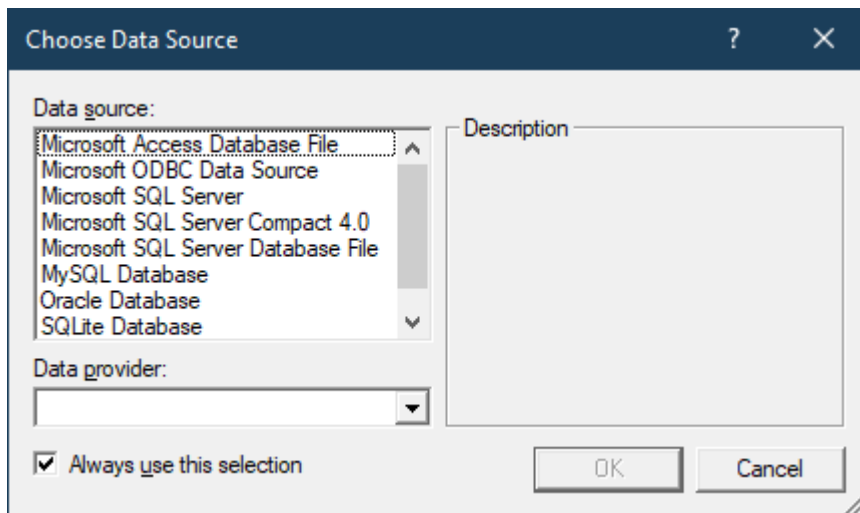
Please also refer to this website for DB connection strings using this link: "DevExpress Connection String" .

Database Engine	DOT.Net	DOT.Net Core
Advantage	<input checked="" type="checkbox"/>	
ASA	<input checked="" type="checkbox"/>	

ASE	<input checked="" type="checkbox"/>	
DB2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Firebird	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MSAccess (32bit Systems)	<input checked="" type="checkbox"/>	
MSAccess (64bit Systems)	<input checked="" type="checkbox"/>	
MSSqlServer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MSSqlServerCE	<input checked="" type="checkbox"/>	
MySql	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Oracle	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Pervasive	<input checked="" type="checkbox"/>	
PostgreSQL	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SQLite	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
VistaDB	<input checked="" type="checkbox"/>	

## .NET Framework Connection Strings (Datalogger)

The .NET Framework Provider connection strings are used by the Datalogger resource, and can be defined using the wizard shown in the screen shot below:



This wizard allows fast configuration of the ODBC, Access, SQL Server, MySQL, SQLite and Oracle data sources.



The Connector/.NET 6.9.9 version must be installed when using MySQL.



Oracle support is not available in WebHMI Runtime.

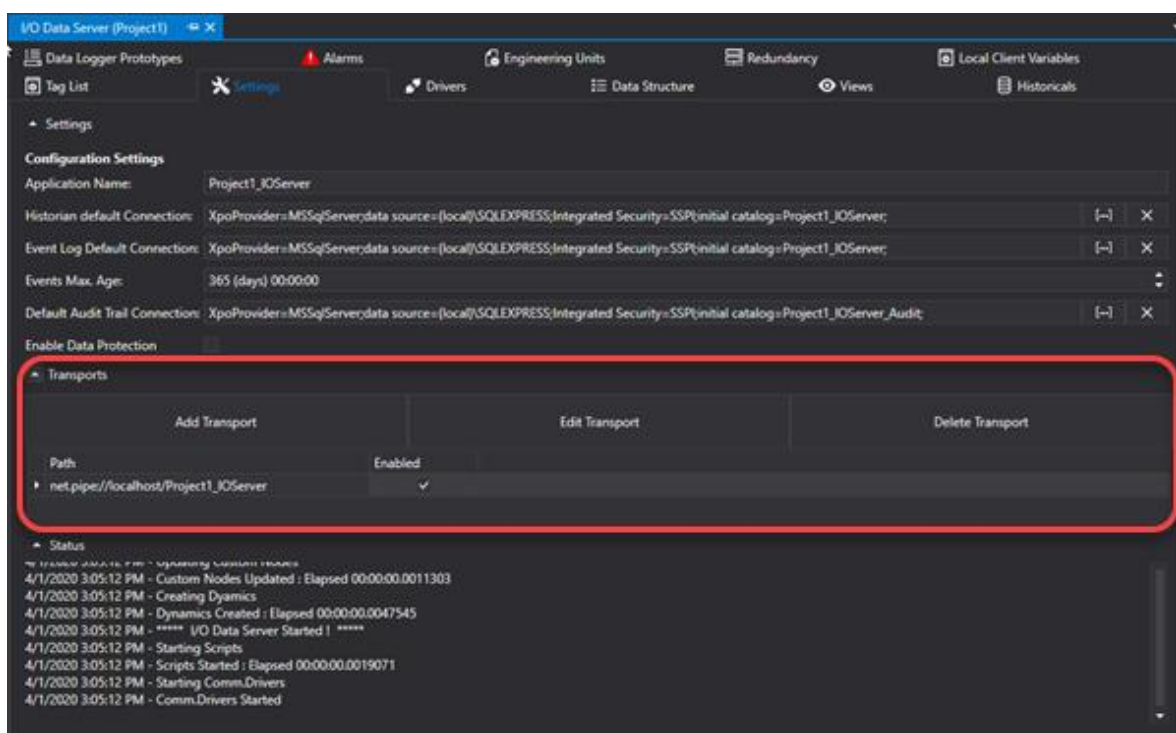
### 4.4.3. Data Transport Definitions

In I/O Data Server settings you will find a window containing a list of Transports which are available for Server project data communications.



This window can then be used for inserting the transport criteria to use for Server and Client data communication. Once the Server has been started up, it will provide the end points according to the transports which have been enabled in the project.

The Clients (any Client) can then use the preferred transport, or one that is the most suitable for the application type being used, independently from the data model being used.



By using the Data Server settings window you can enter, delete or edit the transports available to the Data Server so that they can be used by clients wishing to connect.

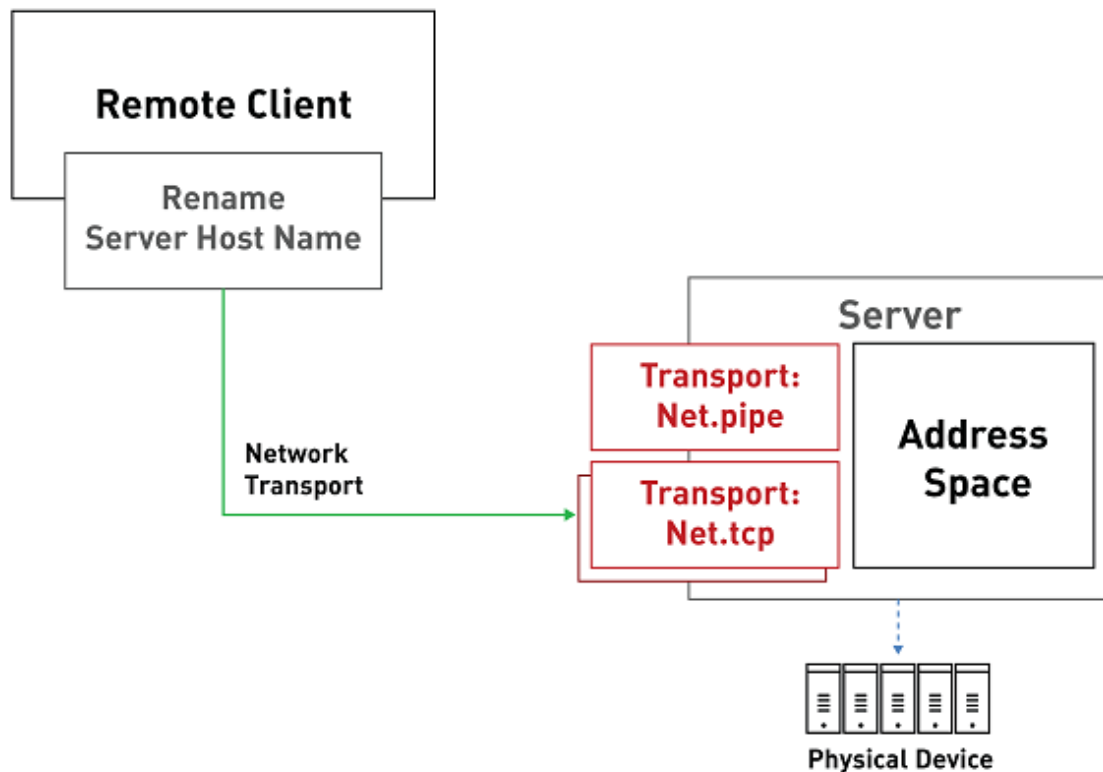
This operations are done using the "Add Transport" or "Edit Transport" window:



The Movicon.NExT Data Server has been specifically designed to make data independent from the Transport type selected to enable distributed Client-Server architectures to use the one most suited to the application type being created.

This window allows you to choose and configure the Transport you wish to add to the Data Server:

- Transport Type
- Server Name
- Port
- Enabled/Disabled



### The local net.pipe transport

Projects are designed with the "net.pipe" transport for default. This transport is the one used in the **local application** for connecting the Client to the Server if not specified otherwise. This means that the visualization Clients assigned with Tags defined in graphical screen objects in the Server will connect using the local net.pipe transport.

This transport is one of the those provided by Microsoft WCF (Windows Communication Foundation) and uses local memory to ensure maximum performances.

**This transport is for 'local' use only and therefore does not consent to network connectivity. If you need to use network connectivity you must enter one of the other transports provided.**



Attention! "net.pipe" is a local transport that entails the Client and the Server run within the same session. This transport cannot be used if the Client project and the Server project run on different machines.

When using a Terminal Server or a Remote Desktop, the net.pipe transport may require user login with an appropriate access level.

### Network Transport

To enable remote Clients to connect to the Server a network transport must be configured in the Server. There are several network transports each one of which differs in aspect according to how they performance or provide security. It is therefore the project engineer to decide which transport to use for their architecture as required or in function with client needs.

The order with which they are used, for instance when a tag connects to a display, is as follows:

- Net.Tcp
- Opc.Tcp
- Https
- Http
- NoSecurityHttp

If only a few of the transports have been enabled, the first available in the project will be used. If none of the transport have been configured, the first one in the Server's configuration file will be used.

#### 4.4.4. Advanced Server Settings

There is a series of Advanced Server Settings that can only be accessed by using the Property Window. This window permits expert users to change certain settings that determine how the Data Server functions.

### General

For further information please see the chapter on "General Settings".

#### Server Application Name

Sets the name of application for OPC UA Server identification.

#### Manufacturer Name

Allows you to enter the manufacturer's name (published by the server)

#### Product Name

Allows you to enter the Product name (published by the server)

#### Product URI link

Allows you to set an URI for the product/manufacture (published by the server)

#### Software Version

Allows you to enter the software Version (published by the server)

#### Build Number

Allow you to enter product build number (published by the server)

### Execution

This group of properties concern those settings for the diagnostics and speech functions.

#### Speech Enabled

Enables the speech function for system message notifications.

#### Diagnostic Enabled (Advanced)

Allows you to enable the publication and management of diagnostic information on the server that clients can access.



Activating this option could have an impact on server performance, so it should be used with caution and only if strictly necessary and for debugging purposes. For example, with the option enabled, in projects with a large number of Tags defined in the Server Address Space, the Server startup may take several minutes.

#### Speech Voice Name (Advanced)

Sets the name of the speech engine.

#### **Default Data I/O Sampling Interval (Advanced)**

Used to set the default minimum refresh rate for Tag value notifications from the server to the client. If no refresh time has been set in the client, the server will use this value set in milliseconds.

#### **Max. Session Count (Advanced)**

Used to set the maximum number of concurrent open sessions.

#### **"Slow" Tag Update Time (Advanced)**

This is used to define the slow sampling interval value for Tags that have been set this family in the advanced "Sampling Interval" property from the Execution group as indicated in topic: "TAG Properties"

#### **"Medium" Tag Update Time (Advanced)**

This is used to define the medium sampling interval value for Tags that have been set this family in the advanced "Sampling Interval" property from the Execution group as indicated in topic: "TAG Properties"

#### **"Fast" Tag Update Time (Advanced)**

This is used to define the fast sampling interval value for Tags that have been set this family in the advanced "Sampling Interval" property from the Execution group as indicated in topic: "TAG Properties"

## **Database Settings**

The advanced settings for additional I/O Data Server configuring can be accessed through the I/O Data Server's Properties Window. A major part of the setting parameters have already been set with a default value, which is considered an optimal value to use in most cases, but can be changed according to your specific needs.

#### **Default Historian Connection**

This is used to set the default connection string to which all the resources based on the historians will report.

#### **Default Event Log Connection**

Sets the predefined connection to be used by all the resources based on the Event log.

#### **Max. Event Age**

This is used to set the max. length of time that data can remain in the Event Log.

#### **Max. Concurrent Historian Accesses (Advanced)**

This is used to set the maximum number of concurrent accesses to the server's historian side. Once this limit has been exceeded, subsequent access requests will be suspended until one of the accesses in progress has concluded.

#### **Historian Retry Waiting Time (Advanced)**

This is used to set the time in seconds to wait between one DB connection attempt and the next.

#### **Max. Historian Retries (Advanced)**

This is used to set the maximum number of historian access retries before flushing data in the cache to a local text file.

**Max. Records in Error (Advanced)**

This is used to set the max. number of records in error in the cache. Once this limit is exceeded, all records pending a new database entry attempt will be flushed to a text file in XML format in the local disk where the server is running.

**Min. Records in Cache (Advanced)**

This is used to set the minimum number of records in the cache pending database entry, below which will be flushed to a local text file in XML format.

**Max. Records in Cache (Advanced)**

This is used to set the maximum number of records in the cache pending database entry. Once this limit has been reached, the new records will be entered in the cache predisposed to flush data to local text files in XML format. In the event that this limit is reached in the second cache, the records will be irretrievably lost.

**Max. Record Deletions (Advanced )**

This is used to set the maximum number of records to be deleted from the database upon every maximum database age control check.

**Max. History Deletion Processes (Advanced)**

This is used to set the max. number of historical data deletion processes in the Database to be run simultaneously.

**Max. History Restoration Processes (Advanced)**

This is used to set the max. number of processes to be run simultaneously to restore historical data..

**Max. History Flush File Size (Advanced)**

Maximum size in MBytes of all Flush files. Once this limit has been reached, the oldest files exceeding this value will be permanently removed. This will be traced in the log with this message: "The maximum size of safe files allowed has been exceeded causing the permanent removal of some files from path '{0}' !"

**Max. Alarm Events (Advanced)**

Sets the maximum number of alarm events to be loaded in the buffer.

**Max. History Continuation Points (Advanced)**

Maximum number of continuation points (as above) used for HistoryRead operations.

**Max. Request Age (Advanced)**

Maximum duration of incoming requests. Take note that the oldest requests are deleted.

## Audit

**Audit Trace Default DB Connection**

This property field is used to set a different connection string as an alternative to the one used by the project for default and represented by the Historian Default connection string in the IO Data Server's

general settings. When set it will be defined with an alternative database within which a UFUAAuditDataItem table will be created to trace recordings of those tags being audited.

### **Enable Data File Protection**

This property field is used when wishing to protect audit recordings from tampering, unintentional or intentional modifications of any kind. This is done by inserting a column with encrypted data in the UFUAAuditDataItem table. The information contained in this column will prevent the mishandling of sensitive data recordings without compromising data integrity.

## **Redundancy**

This section contains various configurable redundancy properties and options. For further information on editable properties in this section please refer to the chapter on "Redundancy Settings".

## **OPC UA Server Option**

Parameter settings for sessions to be managed by the server in client-server communications.

### **Min. Publishing Interval**

The minimum time in minutes to publish data supported by the server.

The server resets subscriptions that have a lower interval than this value. The value returned by the server in a subscription indicates when the server will update the value with the field value.

### **Max. Publishing Interval**

Maximum time in minutes to publish data supported by the server.

The server resets subscriptions that have a higher interval than this value. The value returned by the server in a subscription indicates when the server will update the value with the field value.

### **Publishing Resolution**

This field indicates the minimum difference in ms between the supported publishing intervals. This is always used to calculate the subscription interval which is returned by the server in respect to the one requested by the client.

### **Max. Session Timeout (Advanced)**

This is used to set the maximum time in ms with which a Client session will remain open in the absence of Server communication. It is advised to leave this set with the default value.

### **Min. Session Timeout (Advanced)**

This is used to set the minimum time in ms with which a Client session will remain open in the absence of Server communication. It is advised to leave this set with the default value.

### **Max. Session Count (Advanced)**

This is used to set the maximum number of client session that can be opened on the server.

### **Max. Browse Continuation Points (Advanced)**

This used to set the maximum number of continuation points to be used for browsing operations by clients towards the Server.

The "continuation point" is needed to perform a subsequent request to the server to retrieve remaining information. In fact, the amount of information received by the client might be incomplete due to a limited maximum amount of information that the client can receive for each individual interrogation.

### **Max. Subscription Lifetime (Advanced)**

Defines how long subscriptions can remain open without being published by client.

#### **Max. Message Queue Size (Advanced)**

Defines the maximum number of messages saved in the queue for each subscription.

#### **Max. Notification Queue Size (Advanced)**

Defines the maximum number of notifications saved in the queue for each monitored item.

#### **Max. Notification Per Publish (Advanced)**

Defines the maximum number of notifications saved in the queue for each publication.

#### **Min. Metadata Sampling Interval (Advanced)**

Defines the minimum metadata sampling interval.

## **Advanced Database Settings**

When selecting the Data Server and accessing its Property Window you will notice that it has additional settings called 'Advanced'. These settings are used to configure the I/O Data Server. Each single setting parameter has been preset with a default value which in most cases is optimal. However these parameters can be changed according to specific user needs.

#### **Max. Concurrent Historical Access**

This is used to set the maximum number of concurrent accesses to the Data Server's historical. Any accesses exceeding this limit will be put on hold until one of the ongoing accesses has concluded.

#### **Historical Cache Size**

This is used to set the maximum cache size containing failed data entry attempts. Once this value has been exceeded, all data pending a new attempt to be entered on the database will be flushed onto a text file in XML format in the local disk where the Server is running.

#### **Min. Historical Pending Entries**

This is used to set the minimum number of data in the Cache pending Database entry. When the number of data is less than the value set here, data flushed to a predisposed local XML text file can re-enter into the Cache.

#### **Max. Historical Pending Entries**

This is used to set the maximum number of data in the Cache pending Database entry. Once this limit has been reached, the additional data will be flushed to a predisposed local XML text file. If this predisposed text file also reaches its set limit, data will be irretrievably lost.

#### **Max. Historical Deleting Entries**

This is used to set the maximum number of records to be deleted from the database each time a maximum age database control is performed.

#### **Max. History Flush File Size**

Maximum size in MBytes of all Flush files. Once this limit has been reached, the oldest files exceeding this value will be permanently removed. This will be traced in the log with this message: "The maximum size of safe files allowed has been exceeded causing the permanent removal of some files from path '{0}' !"

### Max. History Alarm Branches

Sets the maximum number of alarm events to be loaded in the buffer.

### Max Session Timeout

This sets the maximum time that a Client session can stay open even when not communicating with the Server. It is advised to leave this set with the default value.

### Min. Session Timeout

This sets the minimum time that a Client session can stay open even when not communicating with the Server. It is advised to leave this set with the default value.

### Max. Browse Continuation Points

This is the maximum number of continuation points used for browsing operations by clients towards Server.

The continuation point is necessary for performing subsequent requests to get remaining information. This is useful when the amount of information received by the client may be incomplete due to a limited maximum amount of information the client can receive in one individual interrogation.

### Max. History Continuation Points

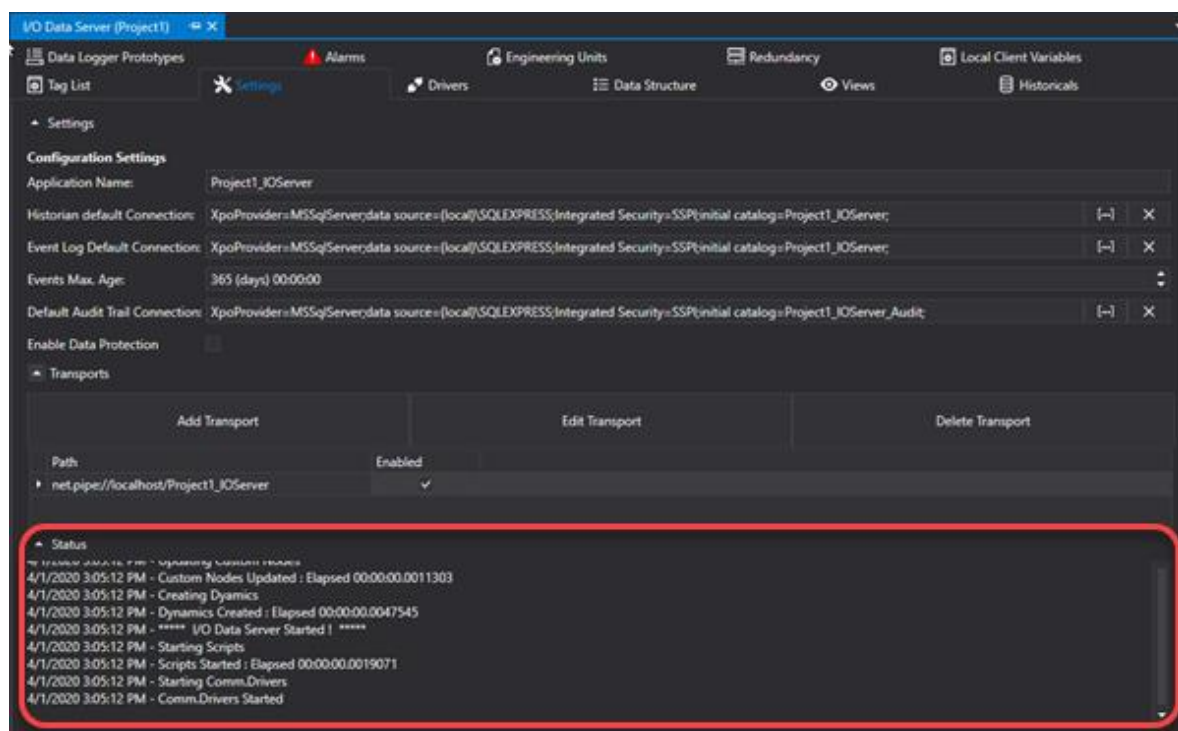
Maximum number of continuation points (as above) used for HistoryRead operations.

### Max. Request Age

Maximum duration of incoming requests. Take note that the oldest requests are deleted.

## 4.4.5. Server Status and Log

There is a window in the I/O Data Server settings that allows the log to be displayed when starting up the I/O Data Server and which activates only when the Server is started up using the "Start Server" command.

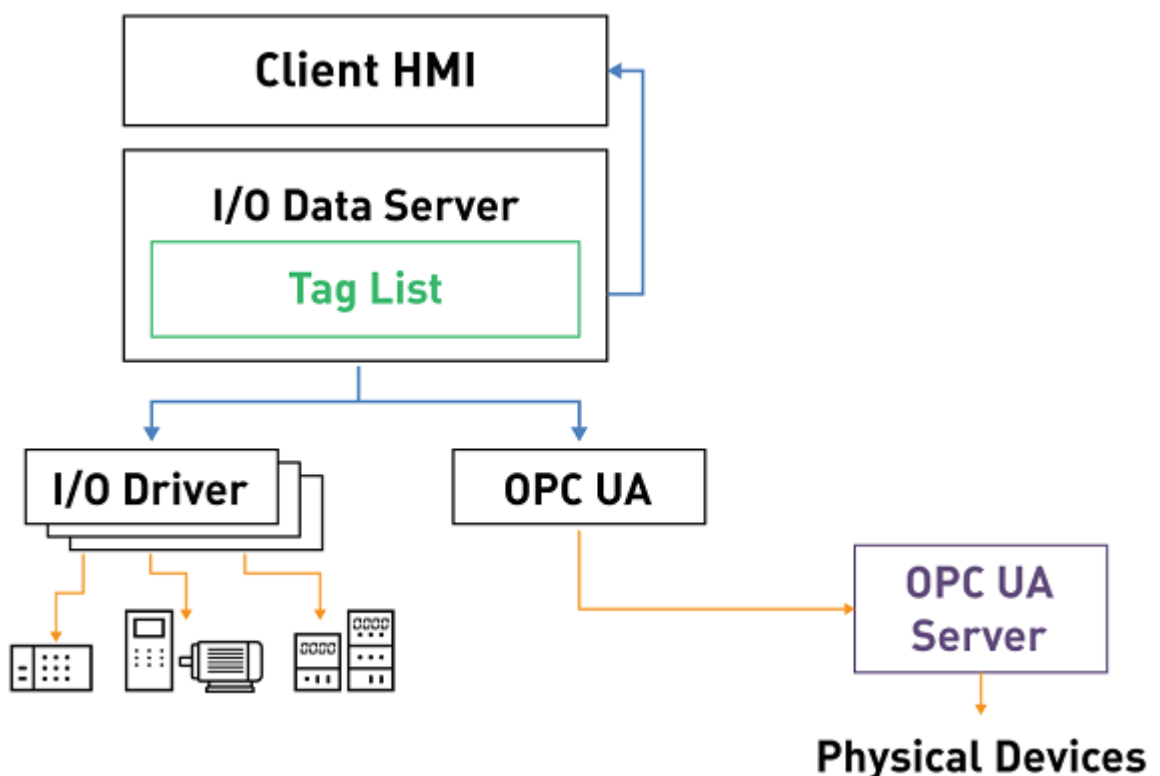




## 4.5. General Driver Concepts

One of the main tasks of the I/O Data Server module is to manage communications with the field devices using different communication channels that manage protocols towards the physical I/Os of external peripherals.

The platform is therefore capable of communicating in real-time with field devices using communication drivers that are included with the product or technology that connects to third party OPC UA Servers. Whatever mode is used all the information collected from the field physical addresses is associated to Tags which are found in the Tag List resource used for the Server's Address Space. The Tag List renders all real-time information visible to all the resources and modules of the entire Automation Movicon.NExT platform whether Server or Client.



This diagram shows an example of an architecture using I/O Drivers or the OPC UA technology to communicate.

The Driver configuration specifications are described in the relating sections of this documentation according to each driver's specific communication protocol type (e.g. Siemens S7. Ethernet-IP, Modbus, Omron and others) along with the OPC UA standard communication technology.



The I/O Data Server Communication Drivers are installed with the same Server. However, additional drivers, or others created by third parties, may be become available after the installation.



Movicon.NExT has been designed to enable expert users to create their own specific communication drivers and integrate them within the platform. However, a SDK is available for developing custom protocols and support and advice is also provided if needed.

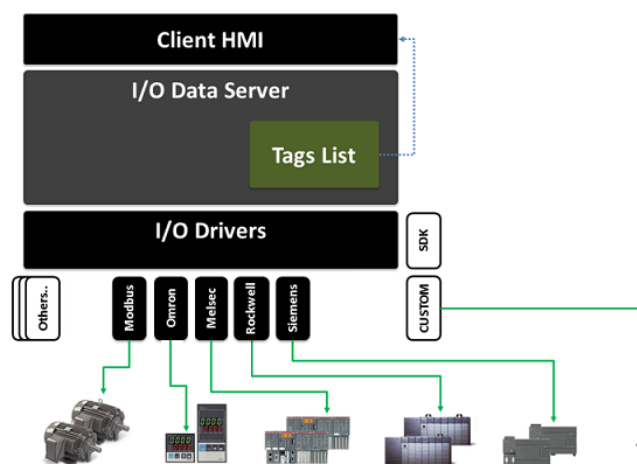
## Drivers

The Communication Drivers are components of the I/O Data Server which allows you to manage specific communication protocols to consent the physical connection to field devices.

They are built form dynamic ".NET" libraries (.DLL file) which, according to exception-based logic, notify information received from device's memory areas connected to the I/O Data Server memory areas and vice-versa, according to the predefined settings for the specific communication protocols. The I/O Data Server, which is the gathering point of dynamic information, places this dynamic information in Tags by means of the address space, to the disposal of all platform modules.

Communication Drivers are available in the I/O Data Server and the list is in continuous expansion. The most commonly used drivers are those for PLC devices (Siemens, Schneider, Rockwell, Omron, Mitsubishi, Saia, Panasonic, etc.), fieldbuses (Profibus, Profinet, Ethernet/IP, EtherCAT, Powerlink, etc.), RTUs and instrumentation (Modbus, etc.) or for infrastructure networks (IEC61870, IEC60850) and civil systems (Konnex, Bacnet, Lon, etc.).

The system's open and modular architecture consents to integration of third party Drivers in the I/O Data Server, to ensure openness needed to the integration of new communication protocols.

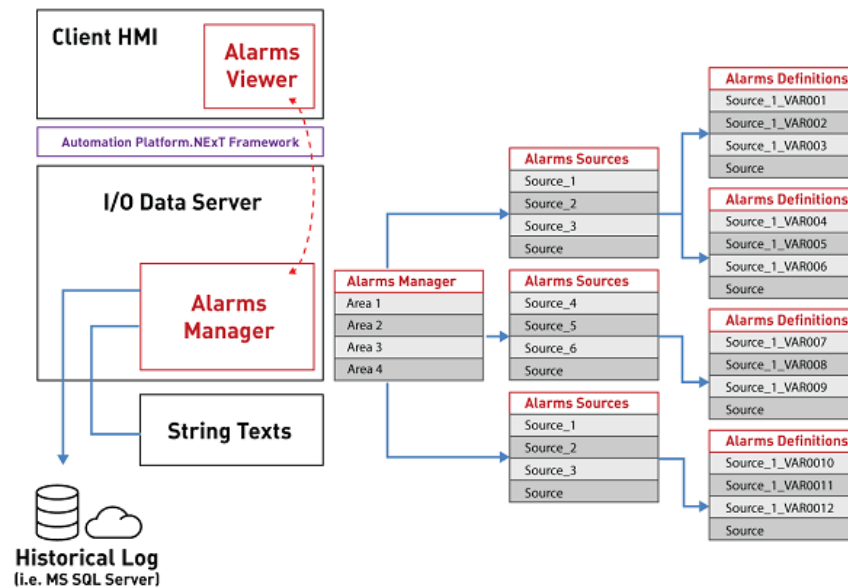


*Generic connection diagram between Movicon.NExT and field devices. The list of available drivers is long and constantly being added to.*

## 4.6. Alarms


### 4.6.1. Introduction to Alarms

The Alarm Server Module allows you to define and manage alarms by first setting their general properties (Type, Thresholds, etc.) in the I/O Data Server 'Alarms' card to then assign them to Tags. In order to get a clear picture of the Alarm concept in Movicon.NExT, look at this diagram:



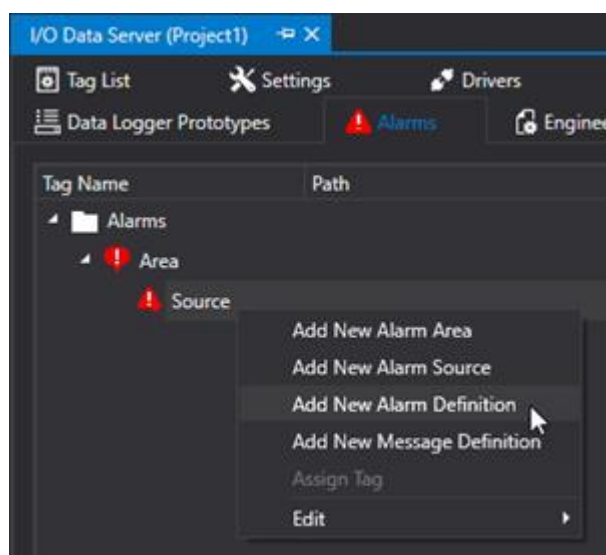
As you can see from this diagram, the Alarm structure provided that possibility to sub-divide alarms in diverse **Areas**, that can be defined diverse **Sources**, each one of which can have their own Alarm **Definitions**.

The alarms defined in the Alarm resource can then be associated to several variables which results very handy in alarm repetitions for example.

 the Alarm management in Movicon.NEXT projects conforms to the OPC UA specifications (standard IEC62541) for Alarm & Conditions typology.

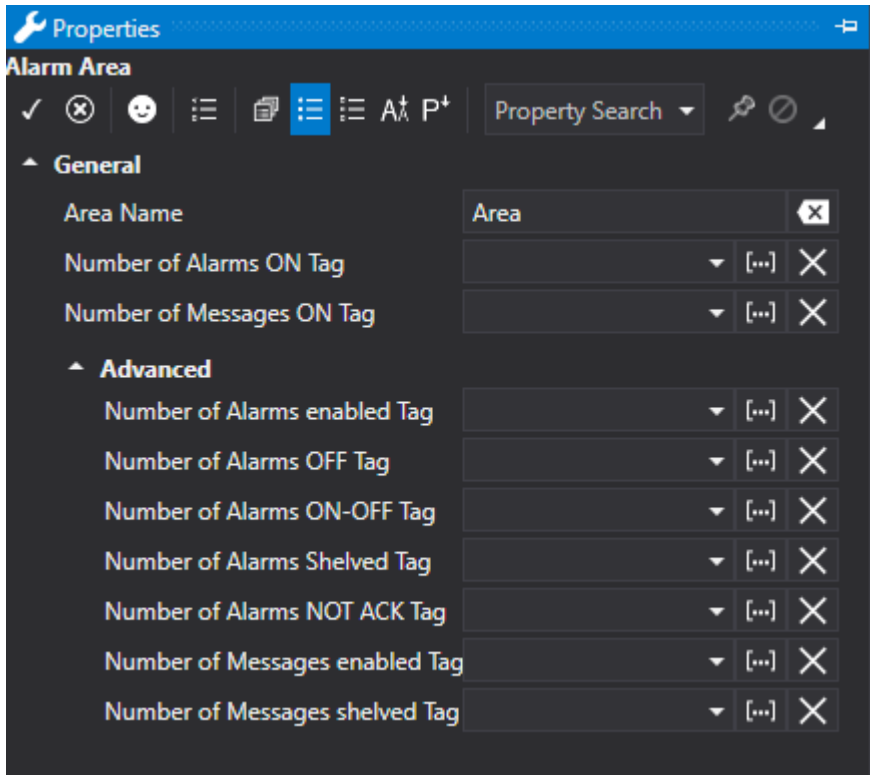
#### 4.6.2. Alarm Definition Settings

An alarm is defined with a structure that has three levels: "Alarm Area", "Alarm Source" and "Alarm Definition".



# Alarm Area

The Alarm Area is used to specify an area which the alarm belongs to in order to manage groups of different alarms. One or more "Alarm Sources" can be inserted in the "Alarm Area" along with other "Alarm Areas".



## General

### Area Name

This is used to set the name for the alarm area.

### Number of Alarms ON Tag

This tag is valued at Runtime with the actual number of active Alarms.

### Number of Messages ON Tag

This tag is valued at Runtime with the actual number of active Messages.

## Advanced

### Number of Alarms Enabled

This tag is valued at Runtime with the actual number of alarms enabled.

### Number of Alarms OFF Tag

This tag is valued at Runtime with the actual number of inactive Alarms.

### Number of Alarms ON-OFF Tag

This tag is valued at Runtime with the actual number of inactive and active Alarms.

### Number of Postponed Alarms Tag

This tag is valued at Runtime with the actual number of postponed Alarms.

### Number of Alarms NOT ACK

This tag is valued at Runtime with the actual number of unacknowledged Alarms.

### Number of Enabled Messages Tag

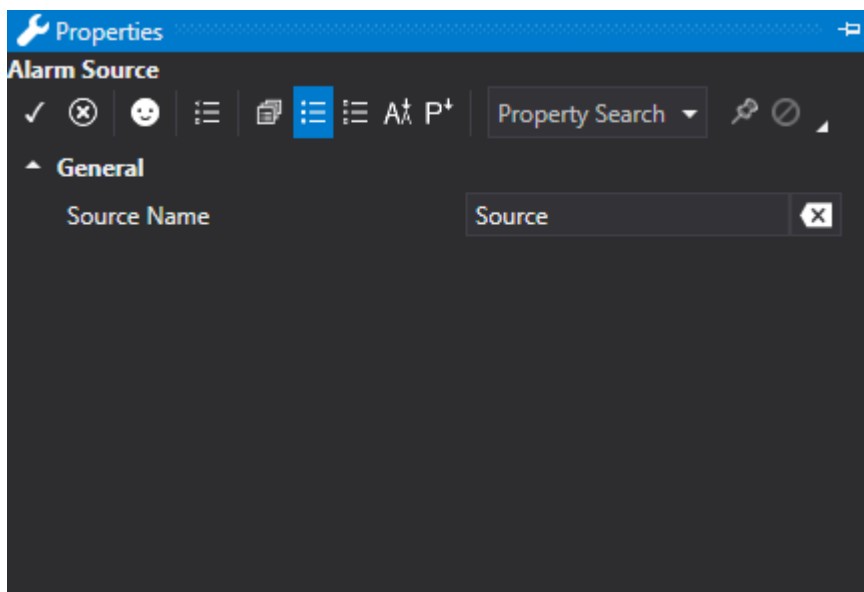
This tag is valued at Runtime with the actual number of enabled Messages.

### Number of Postponed Messages Tag

This tag is valued at Runtime with the actual number of postponed messages.

## Alarm Source

One or more "Alarm Definitions" can be inserted in an "Alarm Source". The Source allows other subdivisions of alarms and a number of Branches can be set and displayed the Source in the Alarm Window.



## Alarm Definition

The "Alarm Definition" represents the actual alarm object in which the various function properties are set. When a new alarm definition is inserted, a configuration window will open where the alarm's different parameters can be inserted. Some of these parameters are present for alarm types, while others are more specific and are displayed based on the "Alarm Type" selected.



## Advanced

### Enable ACK State

This property is necessary for having an acknowledge time in the alarm window (for alarms and messages). This property is also necessary to execute the Reset alarm.

### Enable Confirmed State

This property is necessary for obtaining a reset time in the alarm window. This property is also needed to execute the alarm's Reset command.

### Enable Confirm Statue

This property is necessary for executing the alarm's Reset command.

### Enable Reset State

This property is necessary for suspending an alarm (this function is not yet available in the alarm window).

### Enable Shelving State

This property is necessary to manage the alarm's shelving state.

### Enable Enabled State

This property is necessary for showing the Reason and State in the alarm windows.

### Enable Active State

This property is necessary showing the alarm's change state time in the alarm window.

### Enable Quality State

This property is not yet available.

## Alarm Limits

### Activation Condition

This selection field is only available when the TripAlarm alarm type has been selected. This represents the condition type with which the alarm must be activated in relation to the specified "Activation Value" value. The selection options are:

- **Equals**
- **GreaterThan**
- **LessThen**
- **GreaterThanOrEqual**
- **LessThenOrEqual**
- **NoEqual**

### Activation Value

This selection field is only available when the TripAlarm alarm type has been selected. This represents the reference value to which the alarm must be activated according to the "Activation Condition".

## Execution

### Support ACK

This option is used for enabling the alarm ACK function. When the 'Support ACK' option is enabled and the 'Support Reset' is disabled, the alarm will vanish from the Alarm Window once acknowledged and its activation condition no longer verifiable.

### Support Reset

This option is used for enabling the alarm RESET function. When the 'Support Reset' option is enabled and the 'Support Ack' is disabled, the alarm will not be acknowledged but only reset once its activation condition is no longer verifiable.

### Record Historian Log

This option is used for enabling the Historical Log to record alarms or messages when the relative events occur (ON, OFF, ACK, RESET). These recordings will be kept in the project's "UFUAAuditLogItem" database table created by Movicon.

### Beep

Select this check box if the PC where I/O Data Server is running must emit a Beep, using the system's buzzer, when the alarm activates. This parameter is an Alarm Definition default value that can be changed for each individual "Alarm Threshold" created with the same Alarm Definition. Each 'Alarm Threshold' has its own 'Beep' property that is inherited from the Alarm Definition that can nevertheless be changed.

### Continuous Sound

When checked, this property will play the wav audio file defined in the 'Play Sound File' in a continuous loop.

### Delay Time On

This parameter is used for delaying the alarm's activation. In this case, when setting a value other than zero, in order to activate the alarm, the alarm condition will have to persist the duration of the time set otherwise no alarm event will be generated. This parameter is usually used in those conditions where the Tag value may undergo undesired oscillations that need to be filtered out before a true alarm event can be established.

### Delay Time Off

This parameter is used for delaying the alarm's deactivation. In this case, when a value other than zero is specified, in order to deactivate the alarm, the alarm condition will not have to persist the duration of the set time otherwise the alarm will remain active. This parameter is usually used in those conditions where the Tag value may undergo undesired oscillations that need to be filtered out before a true alarm event can be established.



The Off time is only used for deactivating the alarm. The On time is always used for changing thresholds such as from HighHigh to High for instance.

## Advanced

### Statistics Data

This is used to associate Tag statistics to the alarm instead of their values.



Possible values are: None, Min, Max, Average, NumUpdates, TotalTimeOn.

Tags to which alarm definitions configured in this property are associated should be enabled with the "General > Advanced: Enable Tag Statistics"

### Enable only if quality is good

This is used to enable/disable the alarm management based on the recording quality. When the variable's quality is not 'good', the alarm will not be managed in any which way whatever.

### Enable Branch History

Activating the "Enable Branch History" will allow all alarm transition events from ON to OFF to be displayed in the Alarm Window.

Practically, in addition to the active alarm event, identified as "Active Branch", the Alarm Window will also display new events which are added for each transition from On to OFF, identified as "History Branch".

In this way, you will be able to have a list of events that will allow you to quickly verify how many times and at what time the alarm was active. In the Alarm Window's "Comment" column a message is displayed with the progressive number of Branches.

To delete a Branch from the Alarm Window, simply acknowledge the event (ACK). Even though the alarm, or the active event, has been acknowledged and is in the inactive (OFF) state, it will continue to be displayed when there are other branches to be acknowledged.

### Reproduce Sound File

It is possible to specify the wav audio file to be played by the PC, where the O/I Data server is running, when the alarm activates. In cases where there are several alarms in use, the file linked to the alarm with the highest severity will be played.

### Use Time Stamp

Enabling this will use the data time stamp as the alarm time.

### Deviation Type

This selection field is only available when either one of the ExclusiveDeviation, NonExclusiveDeviation, ExclusiveRateOfChange or NonExclusiveRateOfChange alarm types is selected. This represents the type of deviation that must be used for activating the alarm. This parameter, combined with the set threshold values, calculates the alarm's activation values. The selection options are:

- **AbsoluteValue**
- **PercentOfRange**
- **PercentOfValue**
- **PercentOfUURange**

For further details on these parameters please refer to the paragraphs which describe how the different alarm types function: ExclusiveDeviation, NonExclusiveDeviation, ExclusiveRateOfChange and NonExclusiveRateOfChange.

### High High Level (Enable)

This check box is used for enabling the "High High Level" alarm's intervention threshold. Once the threshold has been enabled it will be possible to enter the activation value by means of using the appropriate "High High Level (Value)" field.

### High High Level (Value)

This entry field becomes available only when either one of the ExclusiveLevel, NonExclusiveLevel, ExclusiveDeviation, NonExclusiveDeviation, ExclusiveRateOfChange or NonExclusiveRateOfChange has

been selected and when the threshold has been enabled by means of using the "High High Level (Enable)" property. This parameter is used for calculating the "High High" threshold value for alarm activation. The value entered in this field is used if different modes according to the selected "Alarm Type" and any eventual "Deviation Type". For further details on these parameters please refer to the paragraphs which describe how the different alarm types function: ExclusiveLevel, NonExclusiveLevel, ExclusiveDeviation, NonExclusiveDeviation, ExclusiveRateOfChange e NonExclusiveRateOfChange.



The usage of the "HighHigh" threshold can be disabled by using the appropriate Check-Box.

### High Level (Enable)

This check box is used for enabling the "High Level" alarm's intervention threshold. Once this threshold has been enabled it will be possible to enter the activation value using the appropriate "High Level (Value)" field.

### High Level (Value)

This entry field is made available only when either one of the ExclusiveLevel, NonExclusiveLevel, ExclusiveDeviation, NonExclusiveDeviation, ExclusiveRateOfChange or NonExclusiveRateOfChange alarm types are selected and when the threshold has been enabled using the appropriate "High Level (Enable)" field. This parameter is used for calculating the "High" threshold value for alarm activation. the value entered in this field is used if different modes according to the selected "Alarm Type" and any eventual "Deviation Type". For further details on these parameters please refer to the paragraphs which describe how the different alarm types function: ExclusiveLevel, NonExclusiveLevel, ExclusiveDeviation, NonExclusiveDeviation, ExclusiveRateOfChange e NonExclusiveRateOfChange.



The usage of the "High" threshold can be disabled by using the appropriate Check-Box.

### Low Level (Enable)

This check box is used for enabling the "Low Level" alarm's intervention threshold. Once this threshold has been enabled it will be possible to insert the activation value using the appropriate "Low Level (Value)" field.

### Low Level (Value)

This entry field is only made available when either one of the ExclusiveLevel, NonExclusiveLevel, ExclusiveDeviation, NonExclusiveDeviation, ExclusiveRateOfChange or NonExclusiveRateOfChange alarm types are selected along with enabling the threshold using the "Low Level (Enable)" property. This parameter is used for calculating the "Low" threshold value for alarm activation. the value entered in this field is used if different modes according to the selected "Alarm Type" and any eventual "Deviation Type". For further details on these parameters please refer to the paragraphs which describe how the different alarm types function: ExclusiveLevel, NonExclusiveLevel, ExclusiveDeviation, NonExclusiveDeviation, ExclusiveRateOfChange e NonExclusiveRateOfChange.



The usage of the "Low" threshold can be disabled by using the appropriate Check-Box.

### Low Low Level (Enable)

This check box is used for enabling the "Low Low Level" alarm's intervention threshold. Once this threshold has been enabled it will be possible to insert the activation value using the appropriate "Low Low Level (Value)" field.

### Low Low Level (Value)

This entry field is only made available when either one of the ExclusiveLevel, NonExclusiveLevel, ExclusiveDeviation, NonExclusiveDeviation, ExclusiveRateOfChange or NonExclusiveRateOfChange alarm types are selected, along with enabling the threshold using the "Low Low Level (Enable)" property. This parameter is used for calculating the "Low Low" threshold value for alarm activation. The value entered in this field is used if different modes according to the selected "Alarm Type" and any eventual "Deviation Type". For further details on these parameters please refer to the paragraphs which describe how the different alarm types function: ExclusiveLevel, NonExclusiveLevel, ExclusiveDeviation, NonExclusiveDeviation, ExclusiveRateOfChange e NonExclusiveRateOfChange.



The usage of the "Low Low" threshold can be disabled by using the appropriate Check-Box.

### Time Unit

This selection field is only available when an ExclusiveRateOfChange or NonExclusiveRateOfChange alarm type is selected. This parameter is used as a sample time range. When the Tag variation reaches higher than the set value within this time range the alarm will be activated. For further details on this parameter please refer to the paragraphs which describe how the different alarm types function: ExclusiveRateOfChange and NonExclusiveRateOfChange.

### Usa il Time Stamp

Normally the Timestamp of an alarm is the date and time of the alarm activation which, with the common industrial automation drivers, differ only by fractions of a second with respect to the date and time of the variation of the value on the PLC. In the remote control field, on the other hand, where the data is collected with longer time intervals, there could be an important deviation between the alarm timestamp and the actual date / time, therefore it is possible to activate this property in such a way that the timestamp of the alarm is equal to the timestamp of the device.

## 4.6.3. Alarm Types

### TripAlarms type

A "TripAlarm" alarm is a simple alarm which has one activation threshold. It is a normal "ON/OFF" alarm which activates or deactivates according to the set value and condition. For example, if the following Value and Condition are set:

**Activation Condition = GreaterThanOrEqualTo**  
**Activation Value = 1**

The alarm will activate when the Tag value is equal to or greater than 1.

For further information please refer to paragraph: "Alarm Definition Settings".

## ExclusiveDeviation and NonExclusiveDeviation Alarm types

A "Deviation" alarm type has four intervention thresholds which can be exclusive or non exclusive.

The Threshold values for this type of alarm have a particular meaning according to the settings in the "**Deviation Type**" parameter. The alarm will behave according to the "Deviation Type" selection as described below. For further information please see the paragraph on "Alarm Definition Settings".

When the alarm is defined as "Exclusive Deviation" type, each intervention threshold will be exclusive, therefore when the "High High" threshold activates the "High" threshold deactivates. When the alarm is defined as "NonExclusive Deviation" type, the intervention threshold will not be exclusive. Therefore when, for example, the "High High" threshold activates so will the "High" threshold. In Runtime the exclusion of an alarm is based on its Status, which is also visible in the "Alarm Window", whether the alarm be exclusive or non exclusive. When an alarm is an Exclusive type, its activation Status will be recorded with the following text:

**HighHigh|...**  
**High|...**  
**Low|...**  
**LowLow|...**

When an alarm is a Non Exclusive, its activation Status will be recorded with the following text:

**HighHighActive|...**  
**HighActive|...**  
**LowActive|...**  
**LowLowActive|...**

#### **Deviation Type = AbsoluteValue**

When the deviation type has an absolute value (Deviation Type = AbsoluteValue), the alarm intervenes only when the Tag value changes to a higher value to the one set in the thresholds. For example, if the four alarm thresholds are set with the following values:

**High High Level = 10**  
**High Level = 5**  
**Low Level = -5**  
**Low Low Level = -10**

The alarm will trigger with the 'High Level' intervention threshold when the variable's value undergoes a variation of at least 6 positive units, for example if its value switches from 30 to 36.

The alarm will trigger with the "Low Level" intervention threshold when the variable value undergoes a variation of at least 6 negative units, for example if its value switches from 30 to 24.

#### **Deviation Type = PercentOfValue**

When the deviation type is set as a percentage of the value (Deviation Type = PercentOfValue) the values set in the thresholds will be considered as percentage values to be applied to the Tag value. The alarm will trigger if the Tag undergoes a variation higher to the Tag's percentage value. For example if the four alarm thresholds are set with the following values:

**High High Level = 100%**  
**High Level = 50%**  
**Low Level = -50%**  
**Low Low Level = -100%**

and the current Tag value is 10, this means that the four intervention threshold will instantly be calculated with the following values:

**High High Threshold = 20**  
**High Threshold = 15**  
**Low Threshold = 5**  
**Low Low Threshold = 0**

the value is established by taking the percentage rate of the Tag value and then adding it to actual Tag value itself. For example:

**High High Threshold = (100% di 10) + 10 = 20**  
**High Threshold = (50% di 10) + 10 = 15**  
**Low Threshold = -(50% di 10) + 10 = 5**  
**Low Low Threshold = -(100% di 10) + 10 = 0**

By following the example described above the alarm will trigger with the "High Level" intervention threshold when the variable value undergoes a variation of at least 6 positive units, for example if its value passes from 10 to 16. The alarm will go off with the "Low Level" intervention threshold when the variable undergoes a variation of at least 6 negative units, for example if its value passes from 10 to 4.



Warning! Each time the Tag changes value, the intervention thresholds will be recalculated based on the new value.

#### **Deviation Type = PercentOfEURange**

When the deviation type is set as the percentage of the scaled value of the engineering unit associated to the Tag (Deviation Type = PercentOfEURange). The values set in the thresholds will be considered as a percentage value to be applied to the scaled value of the engineering unit associated to the Tag. The alarm triggers when the tag variation exceeds the percentage value of the difference of the scaled value limits. For example, if the alarm's four thresholds are set with the following values:

**High High Level= 20%**  
**High Level = 10%**  
**Low Level = -10%**  
**Low Low Level = -20%**

and the Engineering Unit associated to the Tag has the following values:

**Engineering Unit Low Value = -100**  
**Engineering Unit High Value = 100**  
**Low Raw Input Value = -50**  
**High Raw Input Value = 50**

and the actual value of the Tag is 10, means that the four threshold values will at that moment be calculated with the following values:

**High High Threshold = 50**  
**High Threshold = 30**  
**Low Threshold = -10**  
**Low Low Threshold = -30**

The value is obtained by calculating the difference between the minimum and maximum of the scaled value and then adding the value of the actual Tag. For example:

**High High Threshold = (20% di (100 - (-100))) + 10 = 50**

$$\text{High Threshold} = (10\% \text{ di } (100 - (-100))) + 10 = 30$$

$$\text{Low Threshold} = -(10\% \text{ di } (100 - (-100))) + 10 = -10$$

$$\text{Low Low Threshold} = -(20\% \text{ di } (100 - (-100))) + 10 = -30$$

By following the example reported above, the alarm will sound with the "High Level" intervention threshold when the variable's value undergoes a variation of at least 21 units in positive, for example if its value passes from 10 to 31. The alarm will sound with the "Low Level" intervention threshold when the variable's value undergoes a variation of at least 21 units in negative, for example when its value passes from 10 to -11.



Warning! Each time the Tag changes value, the intervention thresholds will be recalculated according to the new value.

### Deviation Type = PercentOfRange

When the deviation type is set as the percentage of the raw value of the Engineering Unit associated to the Tag (Deviation Type = PercentOfRange) the values set in the thresholds will be considered as the percentage value to apply to the raw value of the Engineering Unit associated to the Tag. The alarm triggers when the Tag variation exceeds the percentage value of the difference of the raw value limits. For example, if the four alarm thresholds are set with the following values:

$$\text{High High Level} = 20\%$$

$$\text{High Level} = 10\%$$

$$\text{Low Level} = -10\%$$

$$\text{Low Low Level} = -20\%$$

and the Engineering Unit associated to the Tag has the following values:

$$\text{Engineering Unit Low Value} = -100$$

$$\text{Engineering Unit High Value} = 100$$

$$\text{Low Raw Input Value} = -50$$

$$\text{High Raw Input Value} = 50$$

and the Tag's actual value is 10, means that at that moment the four intervention thresholds will be calculated with the following values:

$$\text{High High Threshold} = 30$$

$$\text{High Threshold} = 20$$

$$\text{Low Threshold} = 0$$

$$\text{Low Low Threshold} = -10$$

the value is obtained by calculating the percentage of the difference between the minimum and maximum of the raw value and then adding the Tag's actual value. For example:

$$\text{High High Threshold} = (20\% \text{ di } (50 - (-50))) + 10 = 30$$

$$\text{High Threshold} = (10\% \text{ di } (50 - (-50))) + 10 = 20$$

$$\text{Low Threshold} = -(10\% \text{ di } (50 - (-50))) + 10 = 0$$

$$\text{Low Low Threshold} = -(20\% \text{ di } (50 - (-50))) + 10 = -10$$

By following the example above the alarm will sound with the "High Level" intervention threshold when the variable's value undergoes a variation of at least 11 units in positive, for example if its value passes from 10 to 21. The alarm will sound with the "Low Level" intervention threshold when the variable value undergoes a variation of at least 11 unit in negative, for example when it's value passes from 10 to -1.



Warning! Each time the Tag changes value, the intervention threshold will be recalculated in function with the new value.

## ExclusiveLevel and NonExclusiveLevel Alarm types

A "Level" alarm type has four intervention thresholds. In this case, the thresholds can be exclusive or not. This type of alarm intervenes only when the Tag value exceeds the set threshold value. For example if the four alarm thresholds are set with the following values:

**High High Level = 10**

**High Level = 5**

**Low Level = -5**

**Low Low Level = -10**

The alarm goes off with the "High Level" intervention threshold when the variable value becomes higher than 5. The alarm will go off (activates) with the "Low Level" intervention when the variable value becomes less than -5.

When the alarm is defined as "Exclusive Level" type, each intervention threshold will be exclusive. Therefore when for example the "High High" threshold activates, the "High" threshold deactivates. When the alarm is defined as "NonExclusive Level" type, the intervention thresholds will not be exclusive. Therefore for example, when the "High High" threshold activates so will the "High" threshold. In Runtime alarm discrimination based on its activation status, which is also visible in the "Alarm Window", despite whether it is exclusive or not. When an alarm is Exclusive type its activation Status will be recorded with the following text:

**HighHigh|...**

**High|...**

**Low|...**

**LowLow|...**

when an alarm is a Non Exclusive type, its activation Status will be recorded with the following text:

**HighHighActive|...**

**HighActive|...**

**LowActive|...**

**LowLowActive|...**

For further information see paragraph on "Alarm Definition Settings".

## ExclusiveRateOfChange and NonExclusiveRateOfChange Alarm types

A "Rate Of Change" alarm type has four intervention thresholds. In this case the thresholds can be exclusive or non exclusive.

For this type of alarm the threshold values obtain a certain significance according to the "**Deviation Type**" parameter. In addition it is also necessary to set a sample time interval using the "Time Unit"

parameter. Based on the "Deviation Type" selection the alarm will behave as described below. For further information please also refer to the paragraph on "Alarm Definition Settings".

When the alarm is defined as "Exclusive Rate Of Change" type, each intervention threshold will be exclusive, therefore when the "High High" threshold activates the "High" threshold deactivates. When the alarm is defined as "NonExclusive Rate Of Change" type, the intervention threshold will not be exclusive. Therefore when, for example, the "High High" threshold activates so will the "High" threshold. In Runtime the discrimination of an alarm is based on its Status, which is also visible in the "Alarm Window", whether the alarm be exclusive or nonexclusive. When an alarm is Exclusive type, its activation Status will be recorded with the following text:

**HighHigh|...**  
**High|...**  
**Low|...**  
**LowLow|...**

When an alarm is Non Exclusive, its activation Status will be recorded with the following text:

**HighHighActive|...**  
**HighActive|...**  
**LowActive|...**  
**LowLowActive|...**

#### **Deviation Type = AbsoluteValue**

When the deviation type has an absolute value (Deviation Type = AbsoluteValue), the alarm intervenes only when the Tag value changes with a variation higher than the value set in the time interval samples in the thresholds specified with the "Time Unit". For example, if the four alarm thresholds and the "Time Unit" are set with the following values:

**High High Level = 10**  
**High Level = 5**  
**Low Level = -5**  
**Low Low Level = -10**

**Time Unit = 5 sec**

The alarm will go off with the "High Level" intervention threshold when the variable value undergoes a variation of at least a positive of 6 units in the range of 5 seconds. The 5 seconds count starts immediately upon the first Tag variation. As soon as the Tag exceeds the value of the defined threshold the alarm will go off. However, if this threshold value is not exceeded within 5 seconds, the counter will reset and restart upon the next Tag variation. By taking the example setting described above, if the Tag value passes from 30 to 36 in less than 5 seconds, the "High Level" threshold will go off. Analogously, the alarm will go off with the "Low Level" intervention threshold when the variable undergoes a variation of at least 6 negative units, for example if its value passes from 30 to 24 within the range of 5 seconds. When a Rate of Change alarm is activated, it will remain active even when its value stops changing. At this point, after the "Time Unit" time has elapsed and acknowledged by the user, its state will set to "Inactive".

#### **Deviation Type = PercentOfValue**

When the deviation type is set as a percentage of the value (Deviation Type = PercentOfValue) the values set in the thresholds will be considered as percentage values to be applied to the Tag value. The alarm will go (activates) off if the Tag undergoes a variation higher to the Tag's percentage value in the sample time



interval specified with the "Time Unit". For example if the four alarm thresholds and the "Time Unit" are set with the following values:

**High High Level = 100**  
**High Level = 50**  
**Low Level = -50**  
**Low Low Level = -100**

**Time Unit = 5 sec**

and the current Tag value is 10, this means that the four intervention thresholds will instantly be calculated with the following values:

**High High Threshold = 20**  
**High Threshold = 15**  
**Low Threshold = 5**  
**Low Low Threshold = 0**

the value is established by taking the percentage rate of the Tag value and then adding it to actual Tag value itself. For example:

**High High Threshold = (100% of 10) + 10 = 20**  
**High Threshold = (50% of 10) + 10 = 15**  
**Low Threshold = -(50% of 10) + 10 = 5**  
**Low Low Threshold = -(100% of 10) + 10 = 0**

By following the example described above the alarm will go off with the "High Level" intervention threshold when the variable value undergoes a variation of at least 6 positive units within the range of 5 seconds. The 5 second count starts immediately upon the first Tag variation. As soon as the Tag value exceeds the defined threshold value, the alarm will go off. However, if this threshold value is not exceeded within 5 seconds, the counter will reset and restart upon the next Tag variation. By taking the example setting described above into consideration, if the Tag value passes from 10 to 16 in less than 5 seconds, the "High Level" threshold will go off. Analogously, the alarm will go off with the "Low Level" intervention threshold when the variable undergoes a variation of at least 6 negative units, for example if its value passes from 10 to 4 in the range of 5 seconds.



Warning! Each time the Tag changes value, the intervention thresholds will be recalculated based on the new value.

When a Rate of Change alarm is activated, it will remain active even when its value stops changing. At this point, after the "Time Unit" time has elapsed and acknowledged by the user, its state will set to "Inactive".

### **Deviation Type = PercentOfEURange**

When the deviation type is set as a percentage of the EU Range value of the Engineering Unit associated to the Tag (Deviation Type = PercentOfEURange) the values set in the thresholds will be considered as the percentage value to be applied to the UE Range of the Engineering Unit associated to the Tag. The alarm goes off if the Tag variation is higher than the percentage value of the limit differences of UE Range in the time interval sample specified with the "Time Unit". For example, if the four alarm thresholds and the "Time Unit" are set with the following values:

**High High Level = 20**  
**High Level = 10**  
**Low Level = -10**  
**Low Low Level = -20**

**Time Unit = 5 sec**

and the Engineering Unit associated to the Tag has the following values:

**EU Range Low = -100**

**EU Range High = 100**

**Instrument Range Low = -50**

**Instrument Range High = 50**

and the actual value of the Tag is 10, means that the four intervention thresholds will be calculated with the following values:

**High High Threshold = 50**

**High Threshold = 30**

**Low Threshold = -10**

**Low Low Threshold = -30**

The value is established by taking the percentage of the difference from the minimum and maximum value of the EU Range and then adding it to the same actual Tag value. For example:

**High High Threshold = (20% di (100 - (-100))) + 10 = 50**

**High Threshold = (10% di (100 - (-100))) + 10 = 30**

**Low Threshold = -(10% di (100 - (-100))) + 10 = -10**

**Low Low Threshold = -(20% di (100 - (-100))) + 10 = -30**

By following the example described above the alarm will go off with the "High Level" intervention threshold when the variable value undergoes a variation of at least 21 positive units within the range of 5 seconds. The 5 second count starts immediately upon the first Tag variation. As soon as the Tag value exceeds the defined threshold value, the alarm will go off. However, if this threshold value is not exceeded within 5 seconds, the counter will reset and restart upon the next Tag variation. By taking the example setting described above into consideration, if the Tag value passes from 10 to 31 in less than 5 seconds, the "High Level" threshold will go off. Analogously, the alarm will go off with the "Low Level" intervention threshold when the variable undergoes a variation of at least 6 negative units, for example if its value passes from 10 to -11 in the range of 5 seconds.



Warning! Each time the Tag changes value, the intervention thresholds will be recalculated based on the new value.

When a Rate of Change alarm is activated, it will remain active even when its value stops changing. At this point, after the "Time Unit" time has elapsed and acknowledged by the user, its state will set to "Inactive".

#### **Deviation Type = PercentOfRange**

When the deviation type is set as a percentage of the Instrument Range value of the Engineering Unit associated to the Tag (Deviation Type = PercentOfEURange) the values set in the thresholds will be considered as the percentage value to be applied to the Instrument Range of the Engineering Unit associated to the Tag. The alarm goes off if the Tag variation is higher than the percentage value of the limit differences of Instrument Range in the time interval sample specified with the "Time Unit". For example, if the four alarm thresholds and the "Time Unit" are set with the following values:

**High High Level = 20**

**High Level = 10**

**Low Level = -10**

**Low Low Level = -20**

**Time Unit = 5 sec**

and the Engineering Unit associated to the Tag has the following values:

**EU Range Low = -100**

**EU Range High= 100**

**Instrument Range Low = -50**

**Instrument Range High = 50**

and the actual Tag value is 10, means that the four thresholds will be calculated with the following values at that moment:

**High High Threshold = 30**

**High Threshold = 20**

**Low Threshold = 0**

**Low Low Threshold = -10**

the value is established by taking the percentage of the difference between the minimum and maximum values of the Instrument Range and adding it to the Tag value. For example:

**High High Threshold = (20% di (50 - (-50))) + 10 = 30**

**High Threshold = (10% di (50 - (-50))) + 10 = 20**

**Low Threshold = -(10% di (50 - (-50))) + 10 = 0**

**Low Low Threshold = -(20% di (50 - (-50))) + 10 = -10**

By following the example above reported, the alarm will go off with the "High Level" intervention threshold when the variable value undergoes a variation of at least 11 positive units, for example if its value passes from 10 to 21. The alarm will go off with "Low Level" when the variable value undergoes a variation of at least 11 negative units, for example if its value passes from 10 to -1.

By following the example described above the alarm will go off with the "High Level" intervention threshold when the variable value undergoes a variation of at least 11 positive units within the range of 5 seconds. The 5 second count starts immediately upon the first Tag variation. As soon as the Tag value exceeds the defined threshold value, the alarm will go off. However, if this threshold value is not exceeded within 5 seconds, the counter will reset and restart upon the next Tag variation. By taking the example setting described above into consideration, if the Tag value passes from 10 to 21 in less than 5 seconds, the "High Level" threshold will go off. Analogously, the alarm will go off with the "Low Level" intervention threshold when the variable undergoes a variation of at least 6 negative units, for example if its value passes from 10 to -1 in the range of 5 seconds.



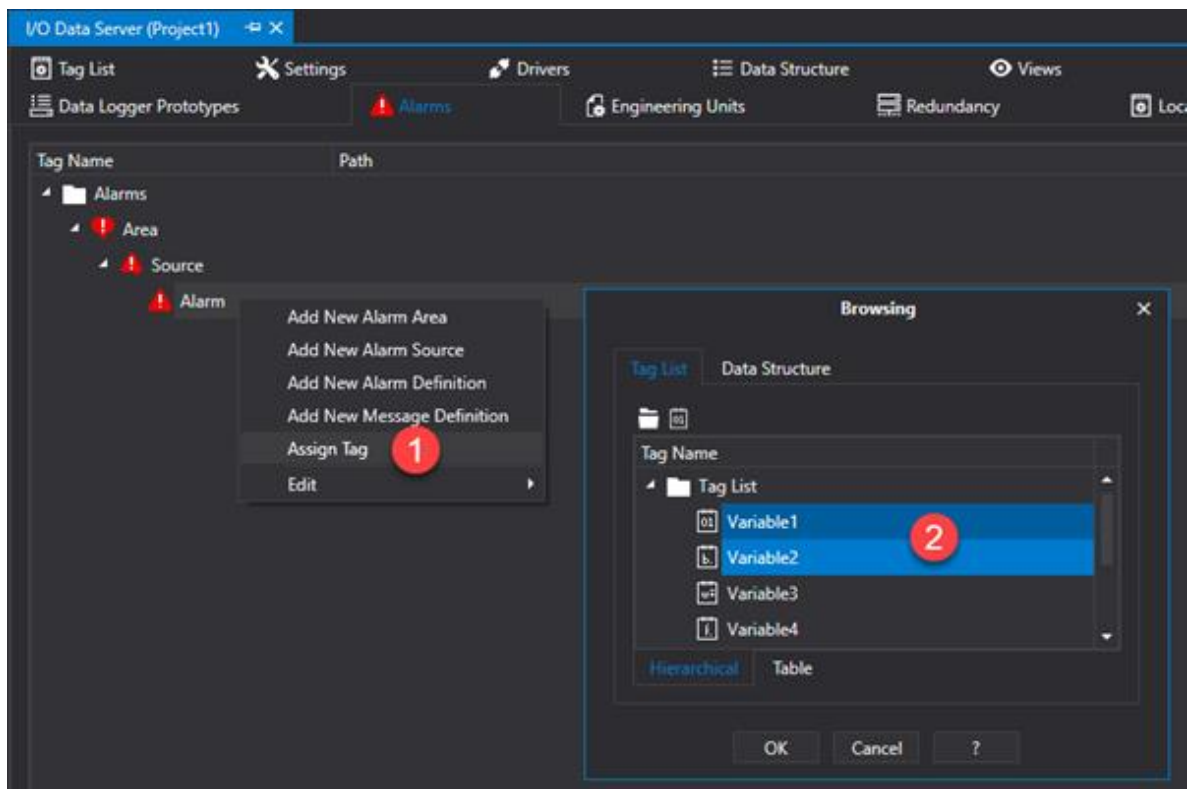
Warning! Each time the Tag changes value, the intervention thresholds will be recalculated based on the new value.

When a Rate of Change alarm is activated, it will remain active even when its value stops changing. At this point, after the "Time Unit" time has elapsed and acknowledged by the user, its state will set to "Inactive".

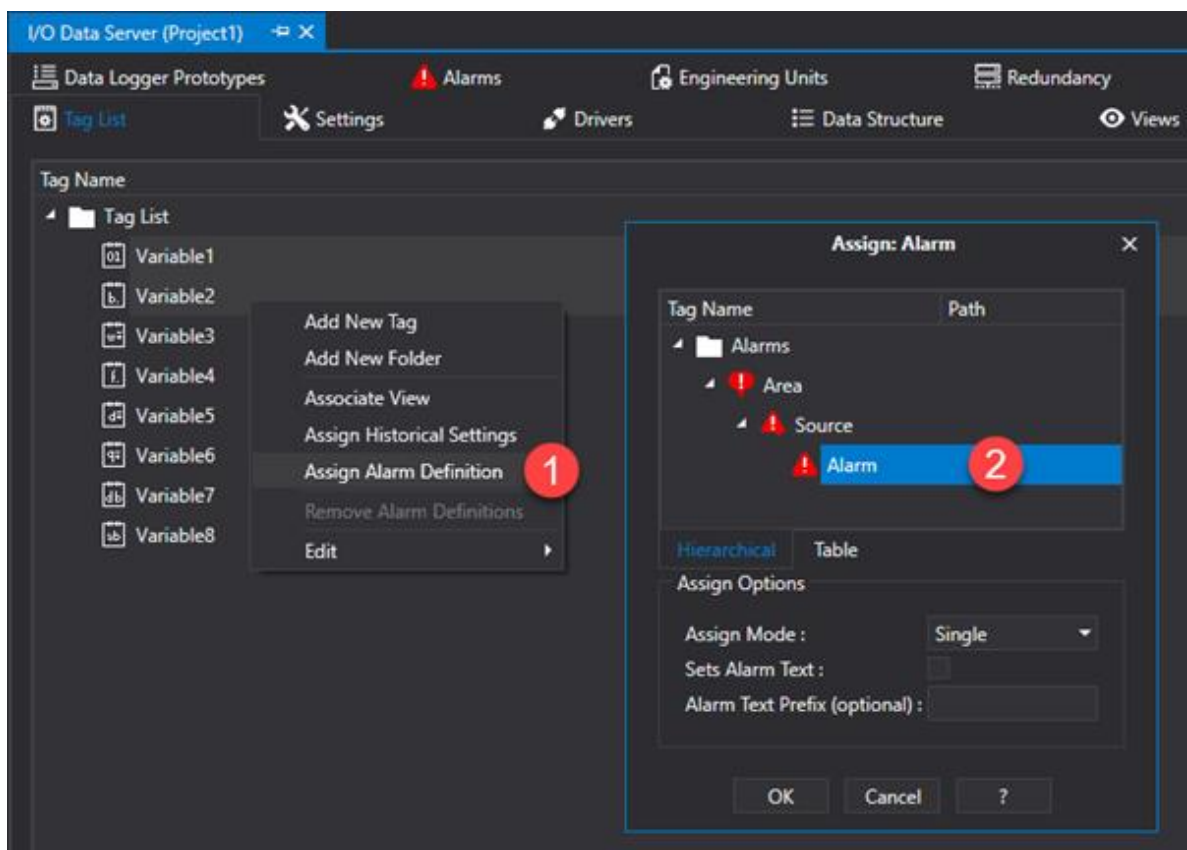
#### 4.6.4. Assigning tags to Alarms

After the Alarms have been defined and configured in the project, they will need to be assigned tags. This operation can be done either from the Alarm resource or the Tag List resource.

From the "Alarm" resource: right click on the alarm and select "Assign Tag". This will open a dialog window allowing you to select one or more tags to associated to the alarm.



From the "Tag List" resource: select one or more tags and, right click to open a contextual menu and select "Assign Alarm Definition". A "Assign Alarm" dialog window will open allowing you to select one or more alarms to the tag.



## Assign Alarm Window

The 'Assign Alarm' window has various options.

## Assign Mode

### "Single"

The alarm is assigned to the entire tag.

### "AnyTagElement"

The alarm is associated to each element of the array type tag, therefore the same number of thresholds will be created as there are array elements.

### "AnyTagBit"

The alarm is associated to each bit of the tag, therefore the same number of thresholds will be created as there are variable bits.

## Set Alarm Text

The 'Set Alarm Text' textbox permits you to set a text to associated to the alarm as a prefix.

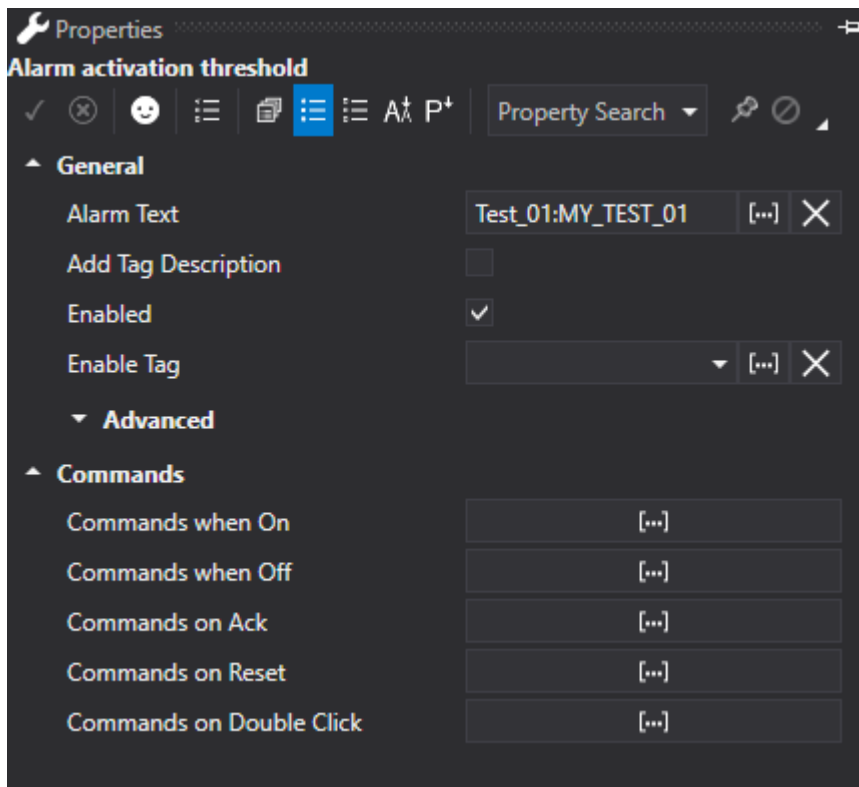
This text will be completed with a incremental numeric '.n' suffix based on the element to which the is associated. l'allarme.

## Alarm text prefix

This is used to define the alarm's text prefix to be used for each new alarm threshold (leave blank to use predefined value).

## Alarm Threshold Properties

Once an Alarm Definition has been associated to the Tags, the alarm associated to each individual Tag can be configured in its properties as indicated below:



## General

In the General Settings window of the alarm threshold section it is possible to define and configure the parameters relating to the management of the most common properties of the alarm thresholds.

### Alarm Text

The text to be shown as the alarm's text is inserted in this field. This text will be displayed in the Alarm Window and will be saved in the project's Historical Log. In cases where this field is left empty, the alarm text will be composed by the name of the Tag followed by the name of the alarm (eg. TagName:AlarmDefinitonName). When a custom text is inserted in this field, it will be displayed within brackets at the side of the alarm threshold's name in the Address Space window.



A static text and a String ID can be inserted in the "Alarm Text" field for managing language changes.

When performing a project language change, the alarm texts already displayed in the Alarm Window will be refreshed with the current language or refreshed by using the Alarm Window's Refresh command or by reloading the Screen.

### Add Tag Description

When this option is selected, the Tag Description will be used as the alarm's text. This function may be very handy when importing Tags from the field devices (e.g. PLC) tag list. The importer also imports by device tag description if any are used to identify the alarm type.

### Enabled

This check-box is used for enabling or disabling the alarm in question.

### Enable Tag

This field is used for selecting a project Tag based on which the alarm is disabled or enabled during Runtime. The alarm is disabled when the Tag is set with the zero value and enabled when the Tag is set with another value that is not zero.

### Activation Tag Value (Advanced)

This edit box is used to insert (or select with the "..." browse button on the right) the name of the variable whose value will be used instead of the one defined in the Alarm Definition's **"Activation Value"** property. This will make the threshold value dynamic so that the variable contents can be modified during Runtime.

When this field is left empty, the contents of the Alarm Definition's "Activation Value" property will be considered as the fixed threshold.

When a Trip alarm type is selected, only one "Activation Tag Value" field will show. When other alarm types are selected, four different "Activation Tag Value" type fields will show, one for each threshold intervention (High-High Limit Tag, High Limit Tag, Low Limit Tag, Low-Low Limit Tag).

### Alias Tag List (Advanced)

One or more variables can be inserted in this field and whose value will be displayed at runtime in the alarm text. The placeholder must be inserted in the alarm text to then be replaced by the Tag's value. The placeholder is defined by the number of tag's position in the Tag list shown within brackets (e.g. {0}). For example if the following text is inserted in the 'Alarm Text' property like so:

**Alarm 1 Active. Minimum value = {0}, maximum value = {1}.**

The two "TagValMin" and "TagValMax" tags are inserted in the Alias Tag List respectively. When the alarm activates in runtime, the placeholder in the text that appears in the Alarm will be replaced by the "TagValMin" tag's value and placeholder {1} will be replaced by the "TagValMax" tag's value .



If a 'String ID' is inserted in the 'Alarm Text' property, any placeholders should to be inserted in the strings of the string table for each defined language.

### Enable only if quality is good (Advanced)

This is used to enable/disable the alarm management based on the recording quality. When the variable's quality is not 'good', the alarm will not be managed.

### Beep (Advanced)

This option is used for setting the alarm beep sound when activated or not. The options are:

- **UseDefault:** selecting this mode the alarm sound will be inherited from the analogic setting in the Alarm Definition with which the threshold was defined with.
- **ForceTrue:** selecting this mode the alarm sound will be executed independently from the analogic setting in the Alarm Definition with which the threshold was defined with.
- **ForceFalse:** selecting this mode the alarm sound will not be executed independently from the analogic setting in the Alarm Definition with which the threshold was defined with.

### Expression (Advanced)

This entry field is used for inserting an expression to enable access to the Tag bit or Array element. For example, the alarm Tag may be a Word type, whereby every single bit corresponds to one single alarm, or an Array type with elements corresponding to different alarms.

The syntax used for accessing the Tag bit is `<.BitNum>`. For example, inserting ".1" in this field means that the alarm will be controlled according to the status of the associated Tag Bit 1. In order to manage more than one alarm bit in the same Tag you will need to assign the same number of alarm templates as there are bits to be managed to the Tag and then specify each of these alarm templates with one of these bits. For example to manage 16 bit of a word you will have to assign 16 alarm templates to the Tag and then assign a different bit in each threshold specified in the Expression.

The syntax for accessing an Array Tag element is `<[element]>`. For example when inserting "[1]" in the field, the alarm will be controlled according to the status of the Array Tag's 1 element. The elements are numbered starting with zero, therefore an Array of 5 elements will be numbered from 0 to 4.

It is also possible to insert an expression to extrapolate a value based on the Tag value associated to the alarm. These expressions must have a syntax like the one used in expressions in objects (see "Expressions in Objects"). An example of this syntax type could be:

`= SQRT([x])`

where the [x] parameter will be replaced with the value of the Tag associated to the alarm. In this case, the alarm's activation value will therefore be the square root of the Tag's value.

## Commands

Below is a list of the commands that can be set for the alarm thresholds:



Commands are always executed within the Client.  
Up to the version 4.0.298 included, commands are not executed by Clients started with the "-start -client" option.

Starting from version 4.0.300 included, the commands are executed by all clients connected to the I/O Data Server, regardless of whether they are local or remote.

#### Commands On

This property is used to associate commands to the alarm threshold to execute when "ON"

#### Commands Off

This property is used to associate commands to the alarm threshold to execute when "OFF"

#### Commands Ack

This property is used to associate commands to the alarm threshold to execute when Acknowledged.

#### Commands Reset

This property is used to associate some commands to the alarm threshold to execute when reset..

#### Commands Double Click

This property is used to associate commands to your alarm threshold to execute when double clicked within the alarm window.

#### Alarm Definition Settings

This button is used for opening the Alarm Definition's setting window with which the threshold was defined.

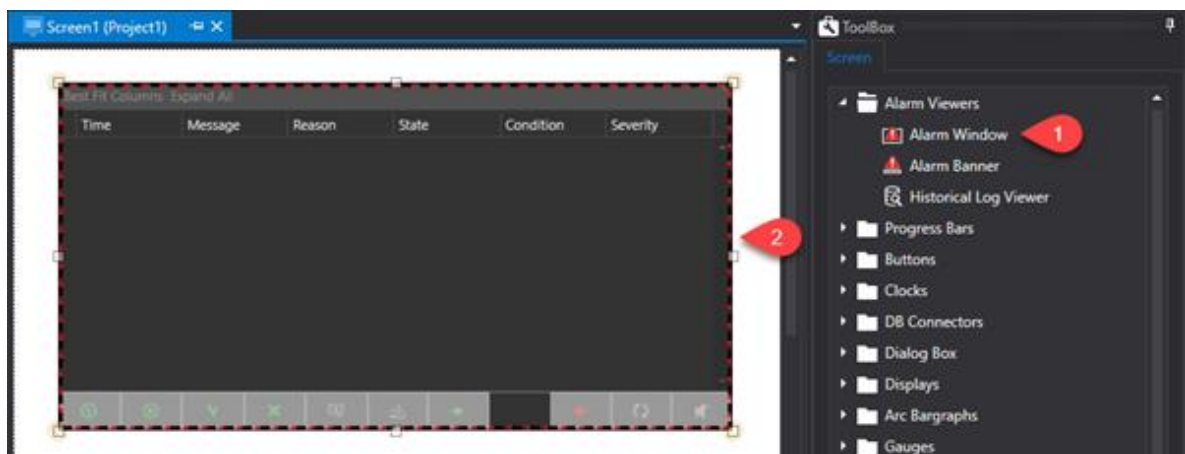
### 4.6.5. Viewing Active Alarms

During the Project Runtime phase the active alarms can be viewed through the "Alarm Viewer" object or the "Alarm Banner" object on screen.



The Server module in your machine will automatically connect to Alarm Viewers inserted on screens.

However, it is also possible to connect the "Alarm Viewer" to a different Server module to the one already in use the running project, for example, to retrieve alarms from one network Server to another project or third party. To achieve this simply specify the Alarm Viewer's reference Server by using the "OPC UA Browser" window in the desired Server and drag the Server onto the Alarm Viewer.





## Alarm Viewer Object

The Alarm Viewer object, available from the Toolbox, is used for listing active project alarms. In addition to viewing alarms the Alarm Viewer can be used for performing other operations by using the buttons described below:

- **ACK:** acknowledges alarm. This event is recorded in the Historical Log table's "Details" column with the "The alarm was acknowledged" message
- **Reset:** resets alarm. This event is recorded in the Historical Log table's "Details" column with the "The alarm was confirmed" message
- **Shelve:** delays alarm activation. The delay time is inserted in the entry field at the side of the button in the "D.hh:mm:ss" format. This event is inserted in the historical log table "Details" column with the "The alarm was shelved" message
- **Unshelve:** the alarm activation delay is suspended. This event is inserted in the historical log table "Details" column as "The alarm was unshelved"
- **Shelve Time:** the shelve time is specified in this field. The time entered in the field will be taken into consideration when an alarm is shelved. Once this time has expired the alarm will be reactivated. Time must be entered in Days, hours, minutes and seconds.
- **Ack All:** acknowledges all the active alarms in the window
- **Reset All:** resets all the active alarms in the window
- **Refresh:** refreshes the window's alarm list
- **Add Comment:** this command opens a dialog window in which it is possible to insert a comment to associate to the selected alarm. This comment will be saved in the alarm's persistence file and then displayed in the Alarm Windows "Comment" column each time the alarm is activated. In addition, this comment will also be recorded in the Historical Log table in the relevant "Comment" column and then displayed in the Historical Log window's "Comment" column. In this case the comment will be shown for all the alarm's different recorded states (active, inactive, acknowledge, confirmed, etc.)
- **Disable Sound:** when managed as a toggle, this button is used for enabling or disabling the alarm sound. This command is linked to the "AlarmSoundState" Server's system variable. Therefore the alarm sound can be toggled directly in this variable ("AlarmSoundState" = True: enables sound, "AlarmSoundState" = False: disables sound). The AlarmSoundState tag's persistence information is recorded in the IsolatedStorage and therefore is UserBased information. If the Windows user is changed, it will also be possible to save the sound state in a different mode.



When the Alarm Viewer Object is used by the **"Web Client HTML5"**, only the "Ack All", "Reset All" and "Refresh" buttons are available for use but not the "Search" tool, scroll bars, column resizer and filter with clauses even though they are still visible.

The data displayed in columns in the Alarm Viewer are:

- **Branch:** indicates alarm occurrences stating whether event is an active event or past event. For further information on displaying Branches please refer to the paragraph on "Alarms Branch".
- **Message:** shows the alarm's text
- **Comment:** shows any comments associated to the alarm
- **Source:** shows the name of the Area and Source belonging to the alarm
- **State:** shows the alarm's status. This information is divided into two parts, the first part defines whether the cause of the alarm is 'Active' or 'Inactive', while the second part shows the which action is still to be performed: 'Unacknowledged', 'Unconfirmed'. For example, an active alarm has not yet been acknowledged and will show as "Active | Unacknowledged" state. On the other hand, an inactive and an already acknowledged and reset alarm will show with the 'Inactive' state. To eliminate the 'Inactive' state messages from window simply use the Refresh command.
- **Condition:** shows the name of the alarm's activation variable

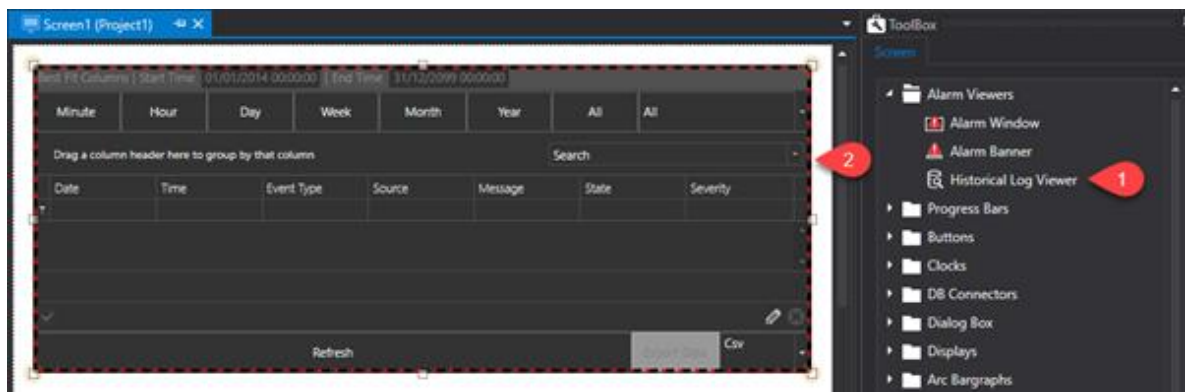
- **Severity:** shows the alarm's severity
- **Time:** shows the date and time of the alarm event

#### 4.6.6. Historical Log Visualization

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

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 and click on the icon to expand group.
3. Select the "**Event Logger Viewer**" object.
4. Drag and drop the selected object to the desired position on screen. To make it simple, we will insert this viewer in the same screen where we previously inserted the Alarm Viewer as shown below:



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

After having inserted the Event Logger Viewer object on screen, you can size and configure its style as needed by using its properties window.



Historical Log viewer provides three tables as previously described for: Alarms, Events, Drivers and System Events. All of them can be viewed by the user at runtime by applying filters or one of the three tables can be forced to display data for viewing. For example, if you want to view the Alarm Log only, use the appropriate property from the object's properties window that enables this to happen.



The Viewer's Refresh command forces access to the DB to refresh the displayed historical data.

In addition, you can use the appropriate viewer object command for applying filters and order bys of recorded data.

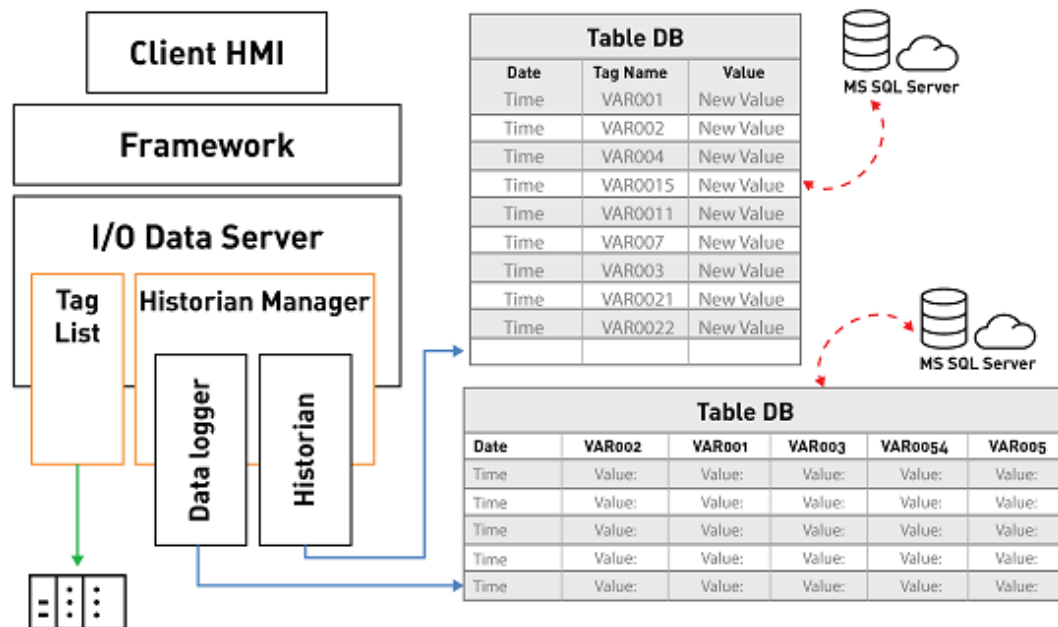
## 4.7. Historian

### 4.7.1. Introduction Historian

The Historian Manager is used to configure the recording of Tags from the I/O Data Server using two different data recording engines:

- Historian
- Data Logger

Both data recording engines record tag values on Database but using two different criteria for doing so which correspond to two different table formats.



## Historians

The Historian records associated Tags on the event of value variations.

For each variable value association, that satisfied the criteria configured in the Historian, a row will be recorded in the Database table containing: the date and time, current variable value, previous variable value and other information.



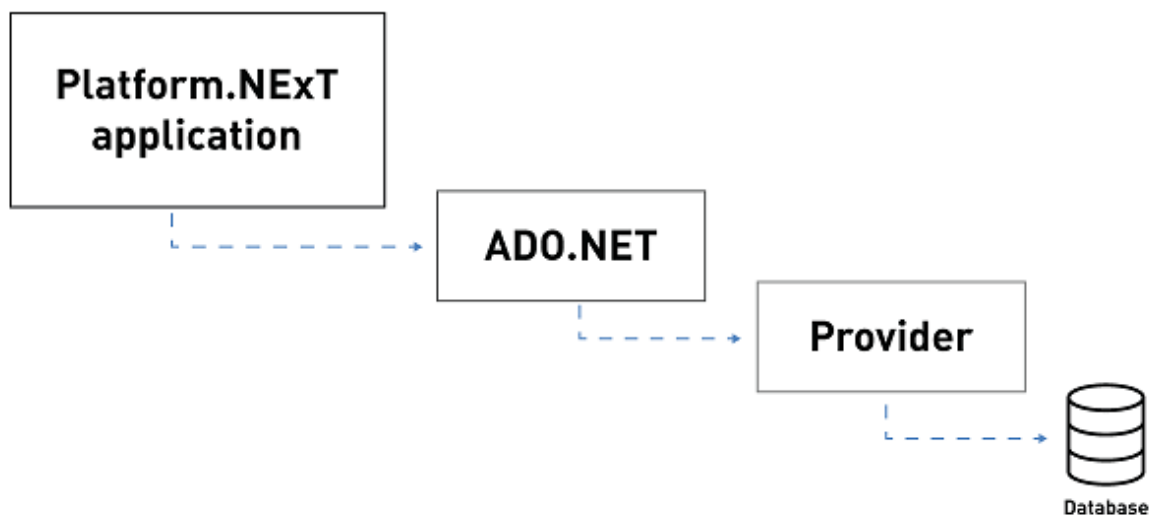
When the Historian has been associated 10 variables that all change their value at the same time, 10 rows will be recorded on the database table, one for each variable.

The use of the Historian is therefore efficient and ideal for keeping track of Tag value changes that do not frequently happen or of those Tags whose trend we cannot predict beforehand and therefore difficult to estimate a fixed recording interval.

## DB Connection

Historians connect to the database by means of the Xpo Provider using the "Historian Default Connection" string for default set in the I/O Data Server's general properties. As an alternative, a different string can be specified for each Historian.

For further information, please refer to the paragraph dedicated to "Xpo Provider connection strings (Historian)" in the topic on "Database Connections".



## Using SQL Server Express as database

When you use the Microsoft SQL Server Express the table generated by Movicon.NexT will be set with the 'Simple' recovery model option.

The log file space in this mode is automatically recovered to limit the disk space requisites and, therefore, to avoid having to manage the space in the transactions log.

The recovery model switches from 'Simple' to 'Complete' only when the 'Enable data protection' for the CFR21 management is enabled.

In this way, all modifications made to the table created by Movicon.NExT will be traced and you will need to instigate a log file backup plan in order to manage the transactions log's disk space. In cases where no backup plan is instigated, there might be a risk that the transactions log file will increase in size to occupy all the space available on the disk.

## Using SQL Server as database

When you use the SQL Server, the tables generated by Movicon.NExT will be set with the 'Complete' recovery model option.

In this way, all modifications made to the table created by Movicon.NExT will be traced and you will, therefore, need to instigate a log backup plan to manage the transactions log's fixed disk space.



If a backup plan is not implemented, the transactions log file might increase in size to then occupy all the fixed disk space available.

## SQL Server CE

As previously mentioned, the standard SQL Server version is installed during the Movicon.NExT setup unlike the Compact Edition of this program which must be downloaded from the Microsoft website using this link:

<https://www.microsoft.com/it-it/download/details.aspx?id=30709>

SQL CE installation is indicated for low performing devices or those which have less memory available. In addition, it provides access protection to data on the database through the use of passwords which can be set using the connection string in accordance with the cf-21 requirements.

Various programs can be used, like the one below, to consult data using external tools:

<http://www.linqpad.net/>

## Data Flush

If the connection to the Database in which data is being recorded is interrupted, the Historian will unload data on to xml files with a name using the following syntax:

**Historical\_YYYYMMddHHMM.FlushedData**

These files are copied in the project's "UFUAServer\Historian" subfolder and then removed after the pending queries have been executed.

## Unavailable DB Connections

In cases in which the connection to the DataBase on which data is being recorded becomes unavailable, the Historian Manager will load the data onto xml files whose names will use the following syntax:

**<HistorianName>\_YYYYMMddHHMM.SafeDataHisLog**

These files are saved in the project's "<ProjectName>\UFUAServer\Historian" subfolder and will be deleted and flushed data saved to the database by executing the pending query.

## Mapping on DB

According to the type of Tag data defined in the DataLogger, a field will be defined in the SQL Server database to contain the Movicon.NExT data. The data correspond to each other as follows:

Movicon	SQL Server
Boolean	bit
Byte	tinyint
SByte	smallint
Int16	smallint
UInt16	int
Int32	int
UInt32	BigInt
Int64	BigInt
UInt64	BigInt
Float	Real

Double	float
String	nvarchar

### 4.7.2. Historian Settings

Each Historian defined in the project has a data recording model type to which the desired Tags can be associated.

Data recording can be in one of these three modes:

- on variation
- timed
- mixed event/timed mode

according to the properties that can be configured as described below.

**Properties**

**Historian Settings**

✓ ✕ ☺ ☹ ☰ ☱ ☲ ☳ ☴ ☵ ☶ ☷ ⌨ ⌂ 🔍 ⌵

Property Search

▲ **General**

Name: Historian ✕

Enabled: ☒

Enable Tag:  [...] ✕

▲ **Database Settings**

DB Connection String:  [...] ✕

Max. Age: 1 (Days) 00:00:00 ⬆ ⬇ ⬆

Table Name:  ✕

▼ **Advanced**

▲ **Execution**

Deviation Type: AbsoluteValue ▼

Deviation Value: 0 ⬆ ⬇ ⬆ ✕

Min. Interval: 0 (Days) 00:00:00.000 ⬆ ⬇ ⬆

Max. Interval: 0 (Days) 00:00:00.000 ⬆ ⬇ ⬆

▲ **OPC UA Historian Settings**

Stepped: ☐

Use Sloped Extrapolation: ☐

▼ **Advanced**

## General

### Name

This field is used for inserting the name of the historian.

### Enabled

This option box allows you to enable the Historian in design mode if disabled (False). In this case, the Server will ignore all disabled Historians (as if they didn't exist) after which it will not be possible to enable them to function during runtime.

This option is enabled for default (True), therefore the Historian results enabled.

### Enabled Tag

This field is used to insert the name of a Tag of the Server to be used in runtime to enable the Historian to record. This Tag will be taken into consideration only when the "Enabled" property is enabled.

If the 'Enabled Tag' is set with the 'zero' value at runtime, the Historian will be disabled and all the Tags connected to it will not be recorded independently from the activated recording mode (on change or time based). Conversely, if the 'Enabled Tag' is set with a value that is different to 'zero' (e.g. 'True', '1', '100', '-10', etc.), the Historian will result enabled and the Tags connected to it will be recorded according to the mode defined.

Note: The use of 'String' or 'Array' variable types are not supported as Historian 'Enabled Tags'. If one of these unsupported variables is inserted, a warning message will show in the Server log at startup and the Historian will ignore the 'Enabled Tag' during runtime but will result enabled for recording.

## Execution

### Deviation Type

This parameter indicates the recording deviation mode. The deviation value used will be the one set in the 'Deviation Value' property above. The Deviation types that can be used are:

- **AbsoluteValue** = a recording is executed according to the variable value's absolute variation meaning when there is a numeric increase or decrease. The condition that determines this is when the absolute variation is ">=" in respect to the set threshold. Cases in which the Historical is associated to a string type variable, the recording will be executed when the string type variable simply changes.
- **PercentOfRange** = a recording is executed according to the value variation percentage taken from the difference between the maximum value (Instrument Range High) and the minimum value (Instrument Range Low) set in the Engineering Unit associated to the variable. The condition that determines this is when the variation percentage is ">=" in respect to the threshold set. This parameter is ignored when the Historical has been associated with a string type variable.
- **PercentOfValue** = a recording is executed according to the current variable's value variation percentage. The condition that determines this is when the variation percentage is ">" in respect to the set threshold. In cases where the threshold is set to 0, this parameter is ignored when the Historical has been associated with a string type variable.
- **PercentOfEURange** = a recording is executed according to the current variable's value variation percentage based on the difference between the maximum value (EU Range High) and the minimum value (EU Range Low) assumed by the variable's associated engineering unit. The determining condition is that the variation percentage should be ">" in respect to the set threshold. This parameter is ignored when the Historical has been associated with a string type variable.

### Deviation Value

This parameter indicates the deviation value that the variable must obtain in order to record. This value may be an 'absolute' type or 'percentOf' type according to the type selected in the 'Deviation Type'

field. When left set at zero, any variable change even if minimal, will activate a recording if the other parameters described below permit it.

### Min. Interval

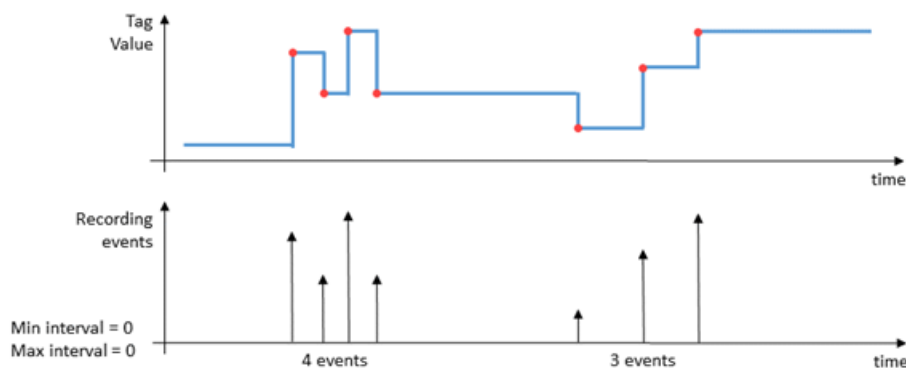
The "**Min. Interval**" represents the time within which each variable variation will not be filtered. Once this time has expired, the variable will be considered and any variation will be recorded. This time count starts after a recording has been executed. Its purpose is to avoid recording very frequent changes that happen one after the other.

### Max. Interval

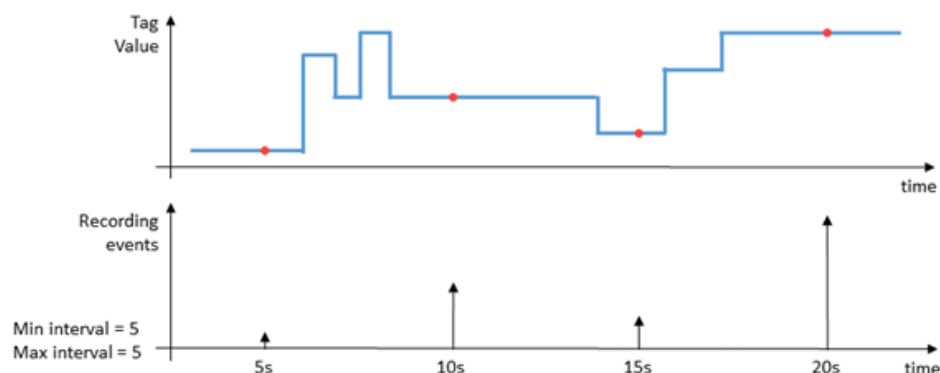
The "**Max. Interval**" represents a time that once expired will prompt a Tag recording independently whether the tag has changed or not. This time determines the fixed tag recording interval time.

**Different modalities can be obtained according to the settings defined for the minimum and maximum recording intervals:**

- Minimum and Maximum interval = zero: RECORDING ON EVENT** In this mode all the tag value variations will be recorded whether they happen very quickly or slowly. Basically, this mode records value variations only when they occur. The ms relating to the RecordDateTime, RecordDateTimeUTC, SourceTimeStamp and ServerTimeStamp columns will also be recorded along with the record on event. Contrary to this, the ms will remain fixed at a constant value.



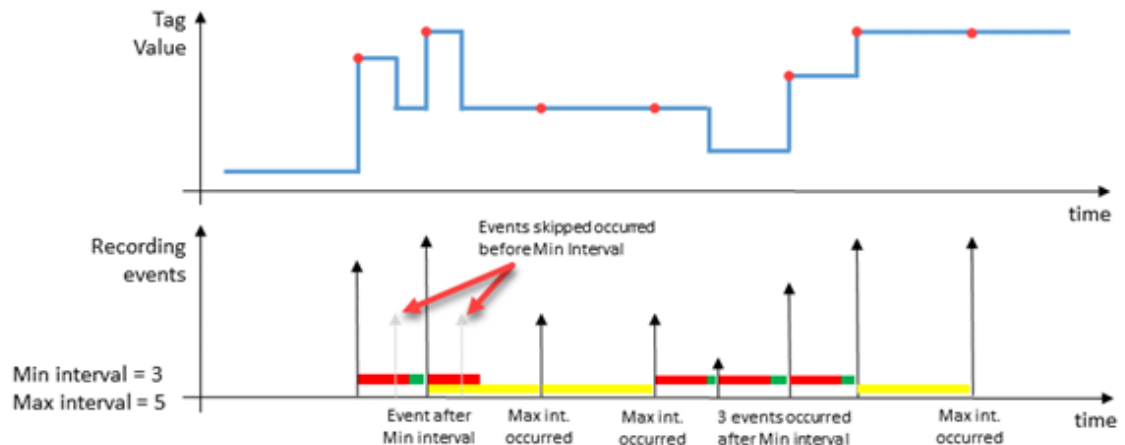
- Minimum and Maximum interval with same value different to zero: RECORDING BY TIME** This mode records by time according to the defined interval time. For example, when setting both values to 3 seconds, a record will be entered on the table at a frequency of every 3 seconds, independently from whether the tag value has changed or not.





- **Minimum interval different than the maximum interval: MIXED EVENT-TIME RECORDING**

Setting two different values in the minimum and maximum interval fields, will cause a mixed behaviour of the previous two conditions. In this case the recordings will be executed when changes occur, according to the 'Deviation Type' property settings, by filtering the fastest changes of the set 'Minimum Time'. In addition, fixed timed recording will also be inserted based on the set 'Maximum Time'.



## Database Settings

### DB Connection String

The "Connection String" parameter is used to define a specific connection to the database for each Historian. When no specific connection has been defined, the historical will use the default connection set in the general "Historian Default Connection" Server connections.

For further information please refer to the paragraph headed "Xpo Provider connecting strings (Historian)" in the topic on 'Database Connections'.

### Max. Age

The "MaxAge:" parameter defines how long data must be kept in the table. The oldest resulting data of the set value will be deleted. Data will not be deleted from this table when setting this parameter with a zero value. The cancellation of data in the DataBase is performed in blocks using the XPO provider so that delete queries with huge quantity of records do not over burden this operation by requiring more time to cancel. Cancellation procedures go into operation after the first Historical log recording if records exceeding maximum age set are detected.

### Table Name

This allows you to use a custom name for the table assigned to the historian. More than one historian can share the same table names.

### Enable Data File Protection (Advanced)

This field is used when needing to protect audit recordings from unauthorized mishandling of any kind whether unintentional or intentional. This is done by inserting a column with encrypted data in the UFUAAuditDataItem object. The information contained in this column impedes the mishandling of sensible data recordings without compromising its integrity.

### **Connection Wait Time (Advanced)**

This parameter represents the waiting time before retrying the data that raised an error.

### **Max Retries (Advanced)**

The "Max Retries:" parameter indicated the maximum number of errors before flushing data on file.

### **Cache Size (Advanced)**

The "Cache Size:" parameter is the maximum cache size for errors which are flushed afterwards.

### **Record Only on Quality Good (Advanced)**

When enabled, this option only allows recordings to be executed only when the communication quality results 'Good' while the tag in question is connected to the field by using a Communication Driver.

## **Historian OPC UA Settings**

### **Stepped**

Keeps the last data point constant or the interpolated value based on a horizontal line adaption.

### **Use Sloped Extrapolation**

This property indicates the mode in which the server executes data interpolation when there are no value limits (e.g. extrapolating the last noted value in future data). A False value indicates that the server will use a SteppedExtrapolation format and will keep the last noted value constant. A True value indicates that the server will transmit the value using the UseSlopedExtrapolation mode. The predefined value is false.

### **Good Data Percentage (Advanced)**

Sets the percentage of good data to be retrieved.

This property indicates the minimum good data percentage within a certain time range, for the StatusCode of the time range specified for the requested processed data to be set to Good. The predefined value is '0'.

### **Bad Data Percentage (Advanced)**

This property indicates the minimum bad data percentage within a given time range requested for the StatusCode in the time range specified for the requested processed data to set to Bad. The predefined value is '0'.

### **Treat Uncertain As Bad (Advanced)**

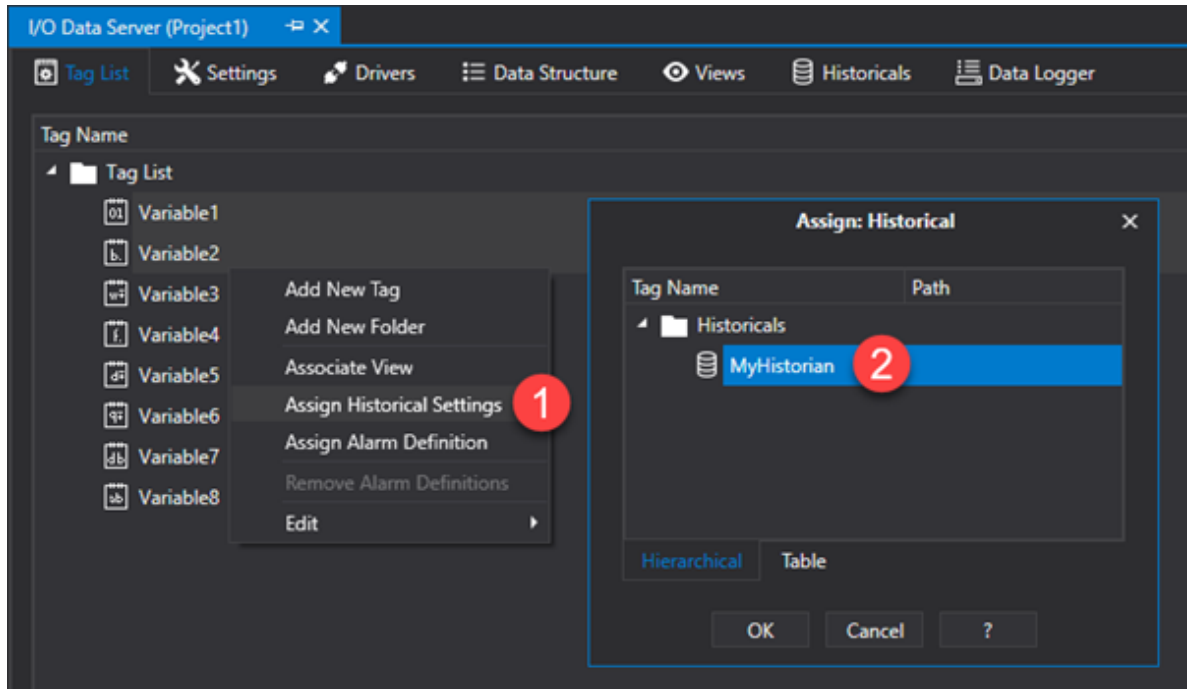
This property indicates in which mode the server treats data returned with Uncertain StatusCode level in respect to the aggregation calculations. A True value indicates that the server will consider uncertainties as Bad, a False value indicates that the server will consider uncertainties as Good, unless the aggregation definition indicates otherwise. The predefined value is False. Note that the value is still considered as uncertain when the statuscode is calculated for the result.

### **4.7.3. Associating Tags to the Historian**

The Historian is only a recording model Template that becomes operative when associated with Tags. There are two ways of associating one or more Tags to a Historian as described below.

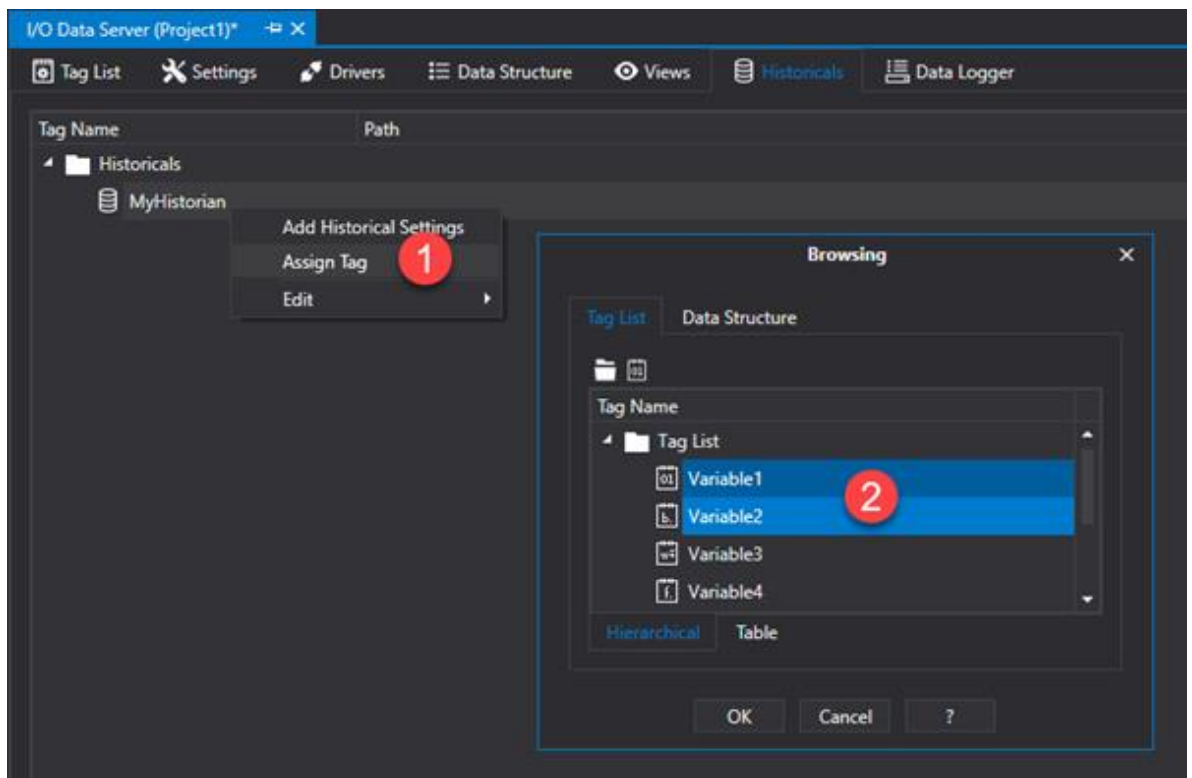
### Assigning a Historian with one or more Tags

To assign a Historian with one or Tags, open the I/O Data Server Tag List and select the Tag or multi-select the Tags that you wish to record. Then select the "Assign to Historian" from the contextual menu to associate the Tag or Tags desired to the previously inserted Historian desired.



### Assigning one or more Tags to a Historian

To assign one or more Tags to a Historian, open the I/O Data Server's Historian resource and select the Historian. Then select the "Assign Tag" from the contextual menu to open the Tag List from which to select the desired tag or tags.



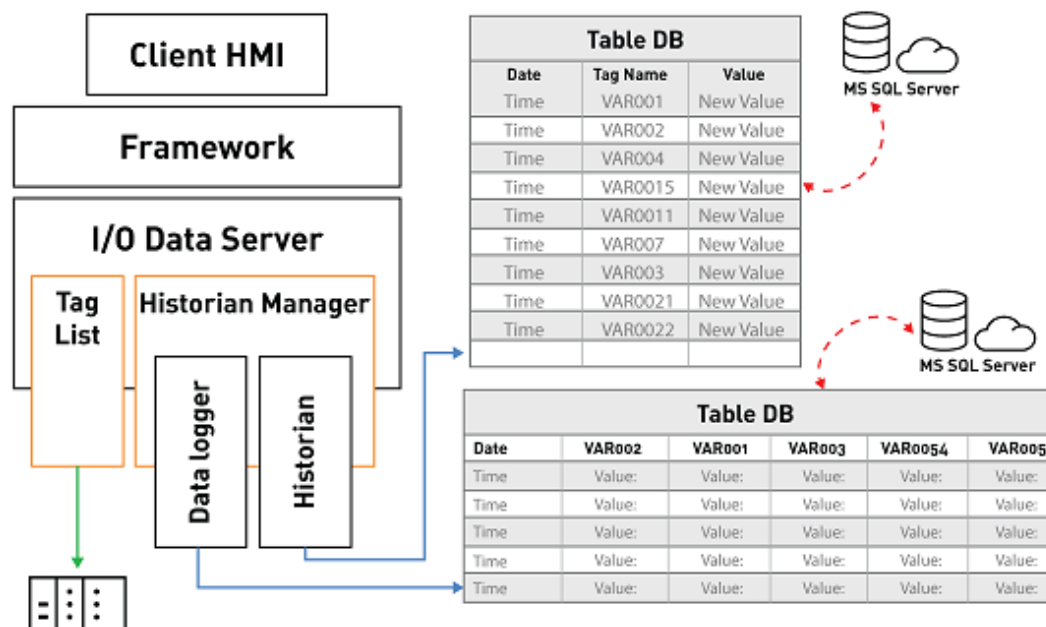
## 4.8. Data Logger

### 4.8.1. Introduction to using DataLoggers

The Historian manager allows you to configure the recording of Tags from the I/O Data Server using two different recording engines:

- Historian
- Data Logger

Both recording models record Tag values on Database but use different table formats for recording data.



### Data Loggers

The Data Logger records the values of all the Tags associated to the same instant independently whether they vary or not.

At each recording instant, which meets the Data Logger criteria, a row will be recorded on the table containing the values of all the tags associated to the Data Logger.



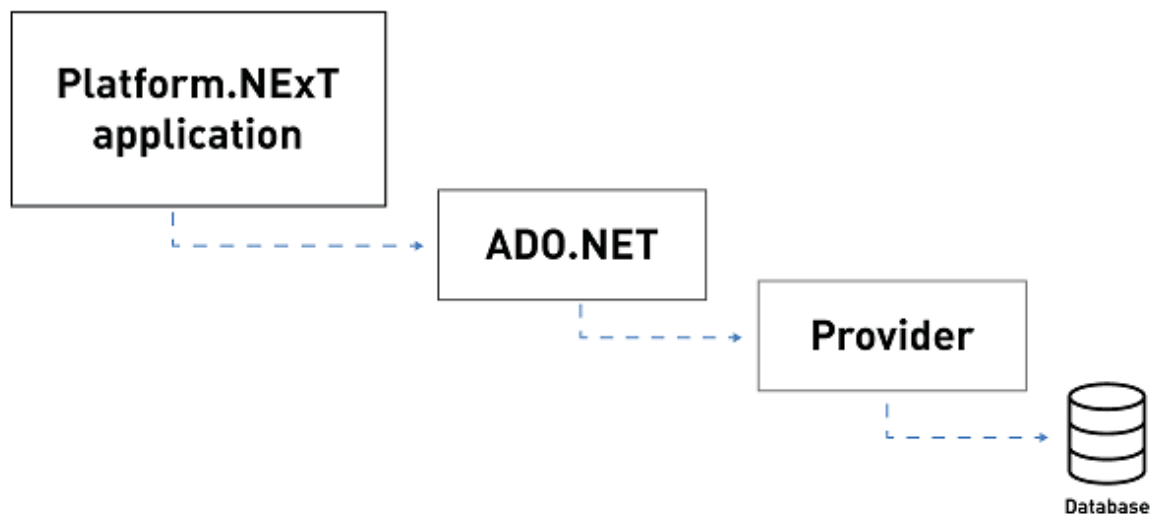
In this case, you will get a table with as many columns as there are associated Tags, with the recording of the values of all the tags included those which have not changed.

Using a Data Logger is therefore ideal for recording the values of multiple Tags periodically and all at the same instance, for example, in order to get a regular clear picture of production trends. However, the Data Logger will record the values of Tags independently of whether they have changed or not and it is important to assess how frequent recordings should be made.

## DB Connection

Datalogger connections to the database takes place through using the .NET Framework Provider. Dataloggers use the "Historian Default Connection" string for default set in the I/O Data Server's general properties (translating it from Xpo to .NET). As an alternative, you can specify a different string for each Datalogger.

For further information, please refer to the paragraph on ".NET Framework Provider Connection Strings (Datalogger)" from the topic on "database connections".



## Using SQL Server Express as database

When you use the Microsoft SQL Server Express the table generated by Movicon.NExT will be set with the 'Simple' recovery model option.

The log file space in this mode is automatically recovered to limit the disk space requisites and, therefore, to avoid having to manage the space in the transactions log.

The recovery model switches from 'Simple' to 'Complete' only when the 'Enable data protection' for the CFR21 management is enabled.

In this way, all modifications made to the table created by Movicon.NExT will be traced and you will need to instigate a log file backup plan in order to manage the transactions log's disk space. In cases where no backup plan is instigated, there might be a risk that the transactions log file will increase in size to occupy all the space available on the disk.

## Using SQL Server as database

When you use the SQL Server, the tables generated by Movicon.NExT will be set with the 'Complete' recovery model option.

In this way, all modifications made to the table created by Movicon.NExT will be traced and you will, therefore, need to instigate a log backup plan to manage the transactions log's fixed disk space.



If a backup plan is not implemented, the transactions log file might increase in size to then occupy all the fixed disk space available.

## Connecting to an unavailable DB

In situations when the connection to the DataBase on which data is being recorded becomes unavailable, the Historian Manager will load the data on xml files whose names will be in the following syntax:

**<DataLoggerName>\_yyyyMMddHHMM.SafeDataLog**

These files, which are saved in the "<ProjectName>\UFUAServer\DataLogger" Project subfolders will then be deleted once communication with the database has been re-established and the flashed data has been saved on database by executing the pending queries.

## Mapping on DB

According to the type of Tag data defined in the DataLogger, a field will be defined in the SQL Server database to contain the Movicon.NExT data. The data correspond to each other as follows:

Movicon	SQL Server
Boolean	bit
Byte	tinyint
SByte	smallint
Int16	smallint
UInt16	int
Int32	int
UInt32	BigInt
Int64	BigInt
UInt64	BigInt
Float	Real
Double	float
String	nvarchar

### 4.8.2. Data Logger Settings

Each Data Logger defined in the project manages a data recording model to which the desired Tags are to be associated. Data recordings can take place:

- timed
- on command

according to the properties that have been configured as described below.

## General

### Name:

Arbitrary name to associate to DataLogger.

### Enable

This option box allows you to enable the Data Logger in design mode if disabled (False). In this case, the Server will ignore all disabled Data Loggers (as if they didn't exist) after which it will not be possible to enable them to function during runtime.

This option is enabled for default (True), therefore the Data Logger results as enabled.

### Enable Tag

This field is used to enter the name of the Server Tag to be used at runtime for enabling the Datalogger recording. This Tag is taken into consideration only when the Data Logger's "Enable" property has been enabled.

When the "Enable Tag" is set to the "zero" value at runtime, the DataLogger will be disabled and all the Tags connected to it will no longer be recorded, independently from the recording mode activated (on command or timed). When the "Enable Tag" is set with a value other than zero (for example, "True", "1", "100", "-10", etc.) il Data Logger will result enabled and the Tags connected to it will be recorded according to the modes defined.



"String" or "Array" tags are not supported as Data Logger "Enable Tags". If one of these tags is inserted, a warning message will be issued in the Server Log at

startup and the DataLogger will ignore the "Enable Tag" property at runtime and will result enabled for recording.

## Execution

### Recording Frequency

This field is used for specifying the time value (gg:h:m:sec) to determine data recording frequency.

### Record on Change

This field is used to enable recording upon the change of at least one of the Tags associated to the Data Logger.

This functionality can also be activated in conjunction with the use of the "Recording Tag" and the "Recording Frequency" fields. When a data Logger has been associated multiple tags, one or more of them can be excluded from the change evaluation by using the "Skip Data Change" property which is available in each Data Logger Tag. For further details please see the topic on "Associating Tags to Dataloggers".

### Record Tag

This field is used to insert the name of a Server tag to be used at runtime to execute Data Logger recordings. However, this Tag is only taken into consideration when the Data Logger's 'Enabled' property has been enabled and that the 'Enabled Tag' property be set to a value different from zero if inserted. If the 'Recording Tag' is set to a value that is different from zero during runtime (e.g. "True", "1", "100", "-10", etc.), the Data Logger will automatically reset the Tag to zero after recording it.



'String' or 'Array' variable types are not supported as Data Logger 'Recording Tags'. If one of these unsupported variables is inserted, a warning message will show in the Server log at startup and the Data Logger will ignore the 'Recording Tag' at runtime.

### Reset Tag

This field is used to insert a Tag name of the Server to be used at runtime to reset data in the Data Logger table. This Tag is only taken into consideration when the Data Logger's 'Enabled' property has been enabled. Any 'Enabled Tag' setting will not influence the reset command which will be executed regardlessly.

If the 'Resetting Tag' is set to a value that is different from 'zero' (e.g. 'True', '1', '100', '-10' etc.), the Data Logger will delete all recorded data in the table up to that point and then reset the Tag to zero automatically.



'String' or 'Array' variable types are not supported as Data Logger 'Resetting Tags'. If one of these unsupported variables is inserted, a warning message will show in the Server log at startup and the Data Logger will ignore the 'Resetting Tag' at runtime.

### Enable Recording Tag

This field is used to insert the name of a Server Tag to be used at runtime to enable the Data Logger to record. This Tag is taken into consideration only when the Data Logger's 'Enabled' property is enabled. If the 'Enable Recording Tag' property is set to the 'zero' value at runtime the Data Logger will not be enabled and all the Tags connected to it will not be recorded independently from the activated recording mode (on command or time based). However if the 'Enabled Tag' is set at a different value that is not zero (e.g. "True", "1", "100", "-10", etc.), the Data Logger will result as enabled and the Tags connected to it will be recorded according to the set mode.





'String' or 'Array' variable types are not supported as Data Logger 'Enable Recording Tags'. If one of these unsupported variables is inserted, a warning message will show in the Server log at startup and the Data Logger will ignore the 'Enable Recording Tag' at runtime and result enabled for recording.

### Hysteresis Time

This field is only available if you enable the "Post on Change" option.

It allows you to specify a time interval, starting from the first variation that occurred, during which any variation of other variables is expected in order to concentrate different variations in a single recording.

The hysteresis time is not considered if the change in value is related to the same Tag, i.e. each change in the same Tag will produce a new registration.

## Database Settings

### DB Connection String

The "DB Connection String" allows you to define a specific database connection for each Datalogger.

When no DB connection is defined, the DataLogger will use the one set for default in the Server's "Historian Default Connection" general property.

For further information, please refer to the paragraph on ".NET Framework Provider connection strings (Datalogger)" from the "database connections" topic.

### Max. Age

This field is used to set the max. time that data can be kept on Database before being eliminated. Data that is older than the value set in this field, in respect to the actual time and date, will be eliminated from the database. The Max. Age of data is always checked when new records are stored.

### Enable Data File Protection

This field is used to choose whether to protect audit recordings from tampering, unintentional or intentional modification of any kind. This is done by inserting an encrypted data column in the UFUAuditDataItem object. The information contained within this column will impede any mishandling of sensible data recordings without compromising data integrity.

### Use Aggregated Tables

Permits the viewing of heightened time periods for frequently sampled data.

When enabled, the stored data procedure produces these tables:

- <DL table name>\_MI for aggregation every minute;
- <DL table name>\_HH for hourly aggregation;
- <DL table name>\_DD for daily aggregation.

For each numeric column; the columns will be created in the aggregated tables with the \_MIN, \_MAX, \_AVG suffix respectively .

- MIN: minimum value reached within the time range in question
- MAX: maximum value reached within the time range in question
- AVG: average value obtained within the time range in question

Criteria used by tables aggregated by the data analysis:

10 min-> MI

720 min-> HH  
14400 min-> DD

This means that data will be taken from the [DataloggerName]\_DD table when exceeding 10 days of extractions; when exceeding 12 hours of extractions (but within 10 days) data will be taken from the [NomeDatalogger]\_HH; when exceeding 10 minutes of extractions (but within 12hours), data will be taken from the [DataloggerName]\_MI; when below 10 minutes of extractions, data will be taken from the [DataloggerName] (the standard DataLogger table).

#### Table Name (Advanced)

The name of the table to use for recording data in is specified in this field. When this field is left blank, the name of the DataLogger set in the "Name" will be used.

#### UTC Time column Name (Advanced)

The name of the column relating to the data in UTC format. The default name will be used when left empty.



If an error should occur while inserting a record, other attempts will be made by using the time interval set in the "Retry wait time" property between each attempt until the number of attempts indicated in the "Max Errors" has been reached. When the number of retries set in the "Max Errors" has been reached, data will be flushed to the xml file indicated in the "Flush path".



When an already existing DataLogger column is renamed, a new column will be created in the table with the new name and the old column will remain untouched.

#### Local Time Column Name (Advanced)

This field is used for specifying the name of the column relating to the data in Local format. When this field is left blank, the default name will be used instead.

#### MSec Column Name

This field is used for entering the name of the column relating to the recording time's milliseconds detail. The default name will be used when this field is left empty.

#### User Column Name

This field is used for specifying the name of the column relating to the user (user who activated recording command). When this field is left blank, the default name will be used instead.

#### Reason Column Name

This field is used for specifying the name of the column relating to the information on what caused the value to change. The default name will be used when this field is left empty.

#### Max. String Size

This field is used to enter the maximum length value of a string to be recorded in the DataBase.

#### Max. Transactions

This field is used to define the maximum number of aggregated operations at one time. In cases with quick recording bit triggers, 10 entries can be performed in a single transaction.

### Max. Errors

This field is used to define the maximum number of errors before flushing.

### Retry Wait Time

This field is used to define the time after which an attempt will be made to perform the operation in the event of an error.

### Max. Cache Size

This field is used to defining the Cache size after which the data will be flushed to file.



If an error should occur while inserting a record, further attempts will be made by using the time interval set in the "Retry wait time" property between each attempt until the number of attempts indicated in the "Max Errors" has been reached. When the number of retries set in the "Max Errors" has been reached, data will be flushed to the xml file indicated in the "Flush path".



In cases where an already existing DataLogger column is renamed, a new column will be created in the table with the new name and the old column will be left untouched.

### Allow Duplicated Rows

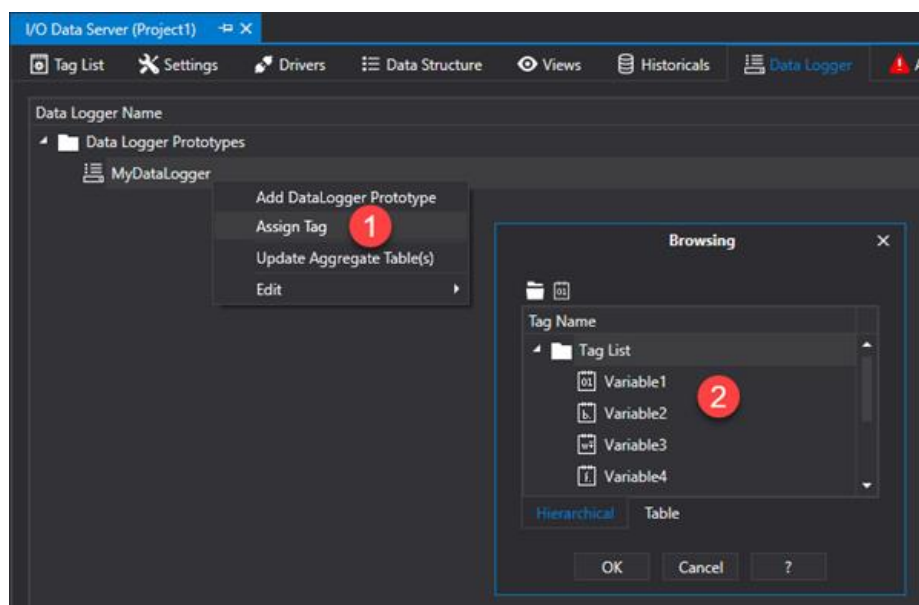
Allows the recording of Data Records with the same Time Data.

### Skip Check Columns Type (Advanced)

This is set false for default. When set true, the DataLoggers columns and tag types implemented at runtime will be checked for consistency.

## 4.8.3. Associating Tags to Data Loggers

A Data Logger is only a recording model template that is made operative by associating it Tags. The association of one or more Tags to a Data Logger is very easy: after having created and configured the Data Logger, select the contextual menu's "Assign Tag" item to open a window from which you can select one or more tags to assign to the Data Logger.





In cases where Tags are added (as columns) after the Data Logger has already performed a recording, the new columns will be added to the Database Table with "null" values for the previous record.

As each associated Tag is represented by a Column in the Data Logger table, each Column can be set with the following properties.

## General Properties

### Column Name

Name of the DataLogger.

### Skip Data Change

When enabled, this Tag will not be considered for record on change.

### Add String Value Column

Adds an additional column to the Data Logger containing the value in string format.

### String Value Column Name

This field allows you to enter the name of the column that will contain the value in string format.

### Tag Colonna

The name of the tag to associated to the Data Logger column is selected from this field.

### Expression

Permits you to define an expression to record the Tag's n-th bit or the array's n-eth element

- to record the n-th bit, use the ".n" expression
- to record the n-th array element, use the "[x]" expression

## Time Stamp Settings

### Add Source Timestamp Suffix

This is used to add the Source Timestamp column (date and time of last tag modification) to the table structure of the selected tag.

### Source Timestamp Column Name

This is used to change the name of the Source Time Stamp column. The name of the column is represented by the name of the tag followed by the indicated column name (e.g. tag002\_col1). When this field is left empty, the name of the column will be taken from the name of the tag plus the "\_SourceTimeStamp" suffix (e.g. tag002\_SourceTimeStamp)

### Add Server TimeStamp Suffix

This field is used to add the Server TimeStamp column to the table structure for the selected tag.

### Server TimeStamp Column Name

This field is used to change the name of the Server TimeStamp column. The column's title is represented by the tag's name followed by the indicated column name (e.g. tag002\_col1). If this field is left empty, the column will be given the name of the tag plus the "\_ServerTimeStamp" suffix (e.g. tag002\_ServerTimeStamp)

## Other Column Settings

### Add Quality Suffix

This field is used to add the Quality column to the table structure for the selected tag.

### Quality Column Name

This field is used to change the name of the Quality column. The column's title will be represented by the name of the tag followed by the name of the indicated column (e.g. tag002\_col1). If this field is left empty, the column will be given the name of the tag plus the "\_QualityTimeStamp" suffix (e.g. tag002\_QualityTimeStamp).

### Add User Name Suffix

This field is used to add the User Column to the table structure for the selected tag.

### User Column Name

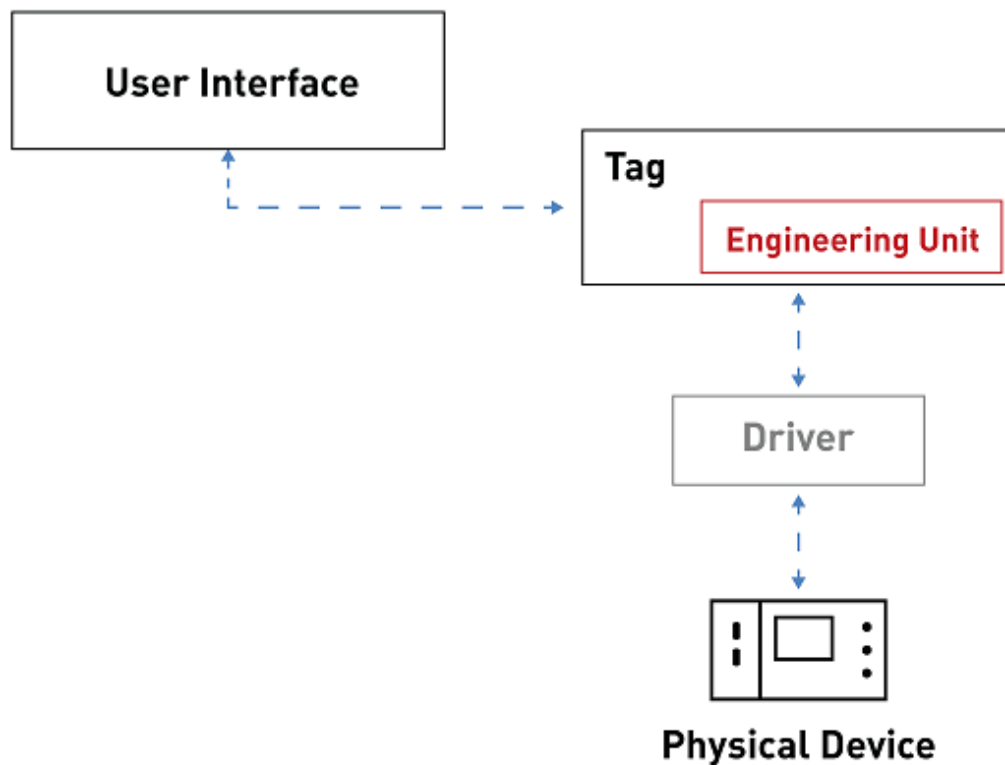
This field is used to modify the name of the User Column, the column's title will be presented with the name of the tag followed by the indicated column name (e.g. tag002\_col1). If this field is left empty, the column will be given the name of the tag plus the "\_UserTimeStamp" suffix (e.g. tag002\_UserTimeStamp).

## 4.9. Engineering Unit

Engineering units are scaled models (aka linear transformation or scaling) that are applied to Tags. By means of using the Engineering Unit, the Tag connected to the field through a communication driver will be scaled according its the settings.

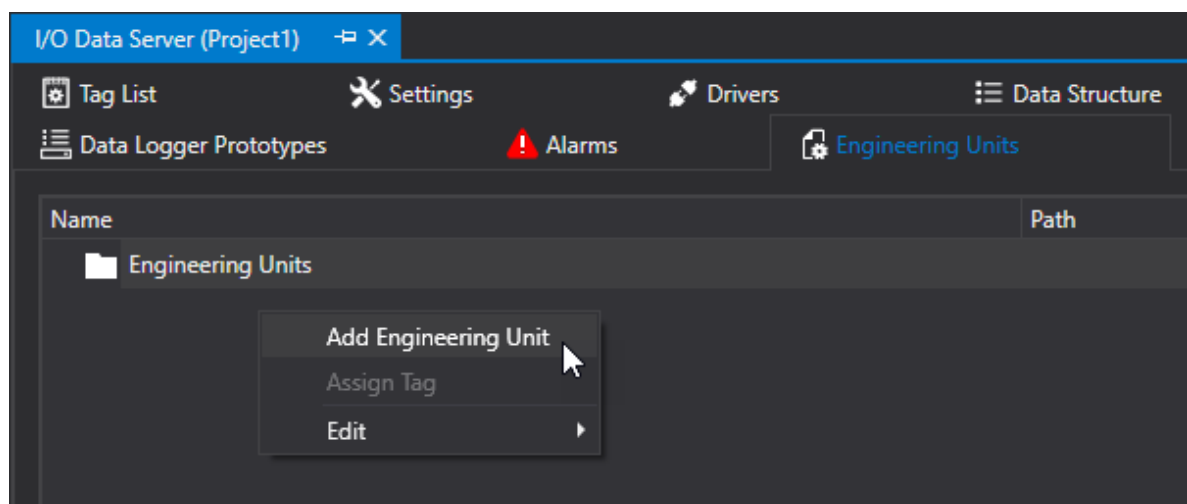
The Raw range is the value range on the device and the Scaled range is the value range in the project's I/O Data Server.

For example, A Tag that has values in the range of 0-1000 on the device and the range of values on the I/O Data Server is 0-100, the formula will be: Tag Value = Device Value/10



Engineering units are useful because they allow the minimum and maximum values of many objects from the toolbox to be configured automatically.

To create a new Engineering Unit simply open the I/O Data Server Engineering Resource and select 'Add Engineering Unit' from the contextual menu.



Once the Engineering Unit has been inserted, you can configure the properties as described below.

## Property of the Engineering Unit

The screenshot shows a 'Properties' window for an 'Engineering Unit'. The window has a blue header bar with a wrench icon and the title 'Properties'. Below the header, the title 'Engineering Unit' is displayed. A toolbar contains icons for undo, redo, delete, copy, paste, and a search icon, followed by a 'Property Search' dropdown menu. The main area is divided into two sections: 'General' and 'Scales'. The 'General' section has two fields: 'Name' with the value 'EngineeringUnit' and 'Unit Measure Label' which is empty. The 'Scales' section has four fields: 'Engineering Unit Low Value' (0), 'Engineering Unit High Value' (100), 'Low Raw Input Value' (0), and 'High Raw Input Value' (100). Each field has a small 'x' icon to its right.

Property	Value
Name	EngineeringUnit
Unit Measure Label	
Engineering Unit Low Value	0
Engineering Unit High Value	100
Low Raw Input Value	0
High Raw Input Value	100

### Engineering Unit Name

Defines the name to assign to the engineering unit.

### Unit Measure Label

This is used to define a text or initials indicating the measure unit and which will be displayed in the objects from the toolbox that will be associated to the Tag that uses the engineering unit.

### Engineering Unit Low Value

This is used to define the minimum value that the Tag will be able to obtain in the I/O Data Server.

### Engineering Unit High Value

This is used to define the maximum value that the Tag will be able to obtain in I/O Data Server.

### Low Raw Input Value

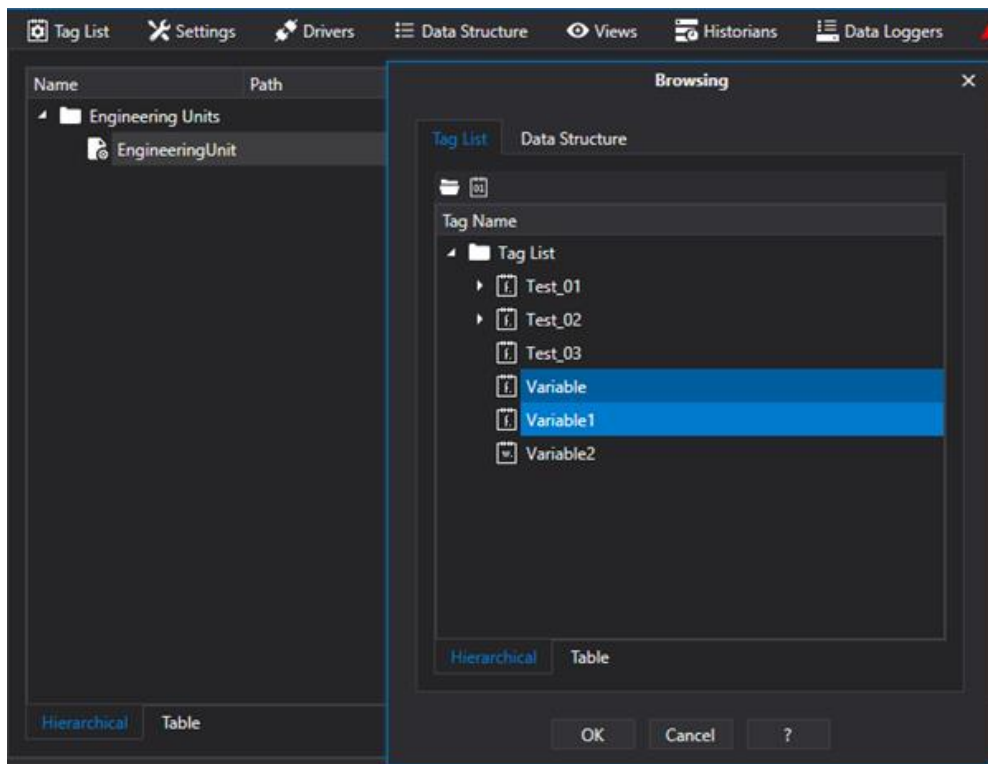
This is used to define the minimum value that the Tag assumes on the Device.

### High Raw Input Value

This is used to define the maximum value that the Tag assumes on the Device.

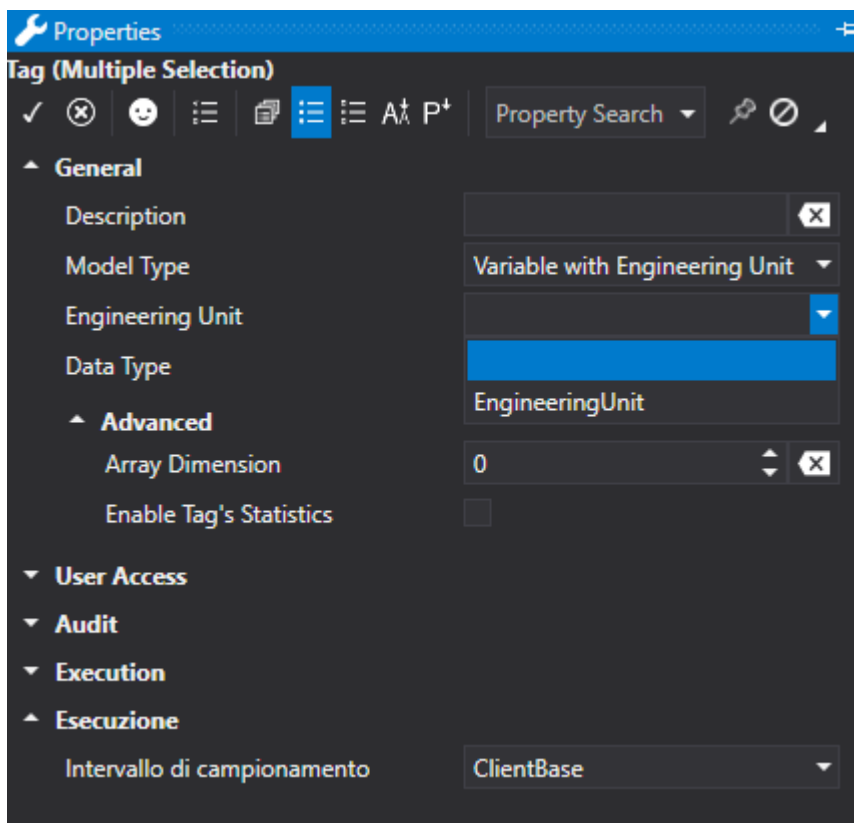
## Associating Engineering Units to Tags

Right click on the Engineering Unit within the 'Engineering Units' resource and then select 'Assign Tag'. This will open a dialog window through which you can select one or more variables to associate to the Engineering Unit.



An Engineering Unit can only be associated to Tags that are "Variable with Engineering Unit" model type or that are not "Bool" or "String" type.

As an alternative, you can use the "Tag List" resource to select one or more variables that are "Variable with Engineering Unit" model types and select the desired resource from the "Engineering Unit" item in the properties window.



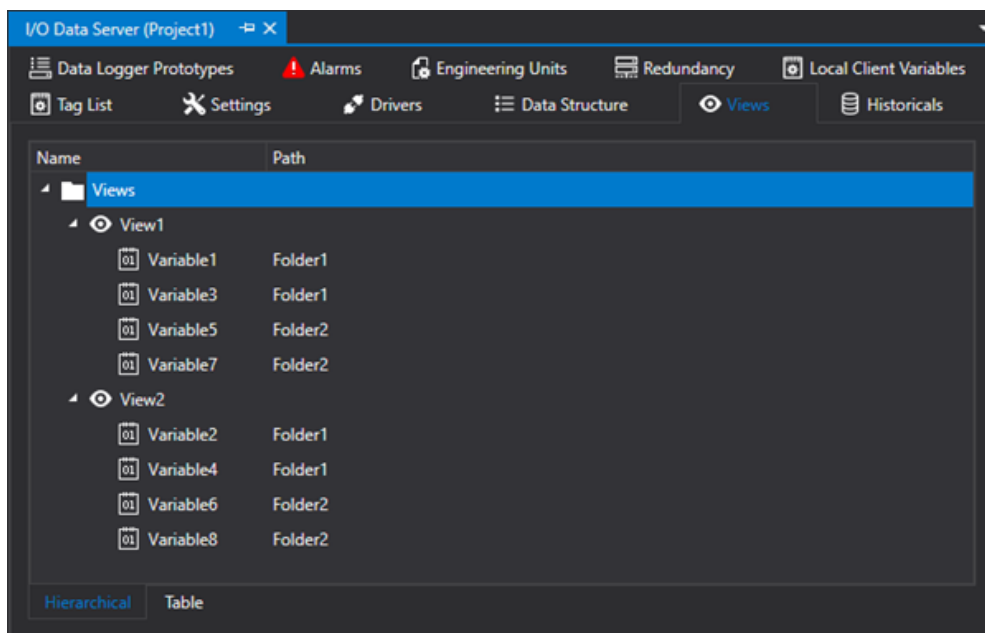


## 4.10. Views

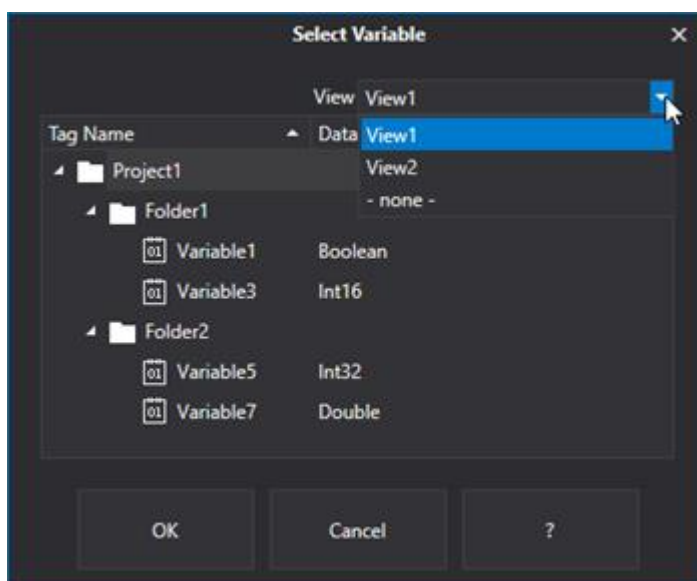
The I/O Data Server's View window is used for inserting and configuring tag group "Views" when selecting tags from the Address Space using the tag browser window.

All tags in the platform are selected with the "browser" window which has been set for default to browse all defined variables.

It is also possible to create one or more "Views" that can then be associated to each individual Tag or Tag Group. This will enable you to display the desired "View" from those inserted in the server by using the "browser" window.



Views are handy to use in the Tag selection window of the Watch Window which can be called at runtime using the SHIFT + F5 combo keys as shown in the screenshot below:



## 4.11. Redundancy

### 4.11.1. Redundancy (Fault Tolerance)

Those control systems which manage critical automation processes must be fault tolerant to ensure reliable and continuous performance and service in the event of system errors or downtimes. Continuous service means full operation of the supervision system and full integrity of recorded process data.

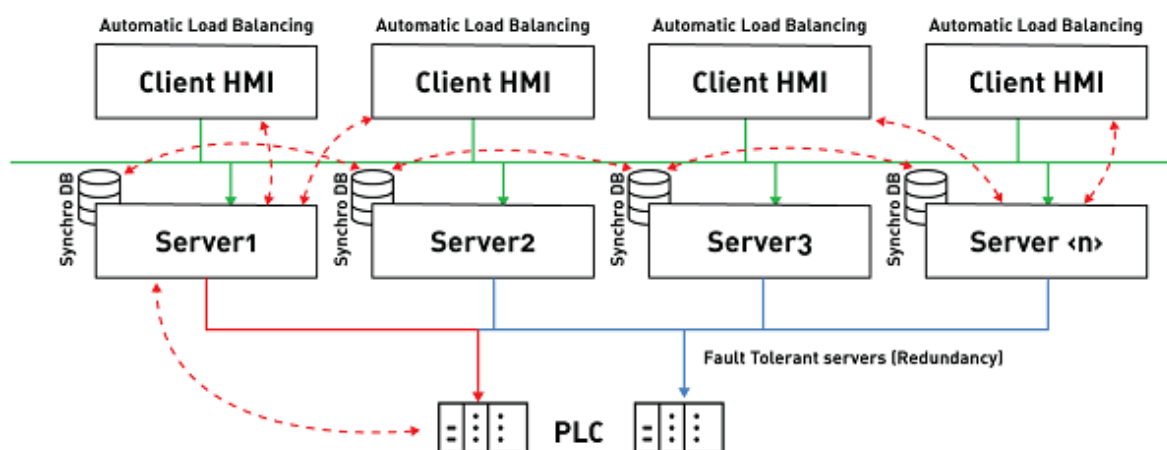
To ensure continuous service the supervision system must provide a 'redundancy' fault tolerant management composed of two or more server systems and two and more client systems that are capable of synchronization with each other automatically.

Movicon.NExT integrates powerful and automatic functions for supporting redundancy management of mission critical server and client stations connected in net. This functionality synchronizes data transparently and automatically from one server to the next in a series of servers connected in net for both communications and historical data. This is done in order to have one Server active as the 'Primary' server with other servers active as 'Secondary' servers continuously at all times.

Analogously, the Client stations are permanently connected to one of the active Servers to ensure an automatic switch over to another server station to avoid interruptions and downtimes.



**The redundancy functions are enabled in runtime only if the Redundancy and Networking options have been enabled on the license of each PC station participating in the system.**



*An example of a fault tolerant architecture for supervision stations. It can consist of several Servers, of which one will be active, and several Clients connected to servers.*

### Redundancy Level

Possessing a redundancy function in automation systems is an absolute must to ensure that in the event of malfunctioning of any kind in one component another identical one is ready on standby to take over automatically.

The "Hot Backup" functions automatically enters the secondary unit into operation without requiring any manual intervention by the operator.

The redundancy concept can be applied both to the hardware as well as the software to determine the minimum loss of data or functionality of the system when switching control over from the Primary unit to the backup Secondary unit.

The redundancy concept in automation systems can be applied to the following **components**:

<b>Field</b>	<b>Connection</b>	<b>Supervision</b>
PLC, Slave, I/O	Serial, Fieldbus, Ethernet network	Server Scada, HMI

The Redundancy functions that have been integrated in Movicon.NExT support the Fault Tolerance functionality in PC stations both on the server and client side so that the communication, visualization and control functions of the active stations can be transferred to the inactive stations completely in automatic. Historical data are also synchronized automatically as well.

The specific Movicon proprietary technology has been designed to synchronize data in an instant no matter even those of large amounts. This is due to synchronizing data acquired while operating in emergency mode by transmitting the data in binary format rather than data structures in database format.

#### 4.11.2. Redundancy Functionality (Fault Tolerant)

The redundancy management with Server and client functionality has been completely integrated in Movicon.NExT platform. It has been explicitly designed to guarantee the automatic intervention of one of the Secondary Servers (inactive) in the event of fault on the Primary Server (active) in such a way that it is completely transparent to the user and detectable after a timeout (modifiable).

In addition, the platform supports the option to define an array of servers to make absolutely sure that the plant system continues to operate if another fault should occur on the active server.

### How the redundancy feature functions

The principle of the Movicon.NExT Redundancy functionality is to have two or more Server modules connected to field devices and connected to each other in an ethernet network. Each one of these Servers must be defined with a project configuration consisting of a list of these Servers connected to each other along with an intervention timeout.

Each one of these servers must also be equipped with a runtime license enabled with the redundancy option.

#### Normal working conditions

During normal working conditions, the Primary Server is connected to the field devices and manages the data and data recording. The other Servers are operative, provide realtime data and record information in the exact same way in perfect synchronization with the Primary Server. They do not communicate with the field even though they are predisposed to do so.

#### Emergency working conditions

In the event of a fault on the Primary Server, the next Secondary Server in line (in a list of backup servers) steps in to communicate directly with the field devices and record data. It will also automatically notify the other Servers on the list that it has taken over as Primary Server. All the other Secondary Servers will adapt accordingly.

If the new Primary Server should go into fault, the next Server in line will assume the role of Primary Server as indicated above.

### Restoring normal working conditions

When a Primary Server returns back into operation it automatically restores the local historical situation and acquires all active information. Once synchronization has taken place the restored primary server will start to communicate with the field devices resuming its primary function. As a consequence the Server, that was active when this happened, will return to its 'secondary' operative condition in accordance to the data provided by its Primary Server.

### Client connections

The data display client stations, connect to the system servers, are permanently and automatically connected to one of the active servers according to the load balancing principle. Therefore, if a Client station is connected to a Server that goes out of service, the Client will automatically search for the next Server station to ensure continuous service in automatic mode.

Thanks to the automatic redundancy functionality based on the working principles described above, the Movicon.NExT supervision system is capable of ensuring excellent service continuity that is ideal for mission critical applications where operation must always be guaranteed in any situation.

## Definitions and Concepts

- **Active Server:** this is the station that in normal working conditions manages the plant, communicates with it, collects data and controls it. If this server should stop working due to a fault in its system, one of the inactive Servers will enter into service.
- **Inactive Servers:** these are the stations (one or more than one) that in normal working conditions permit plant management in redundancy mode by means of sharing variable memory areas. These stations permit independent plant interaction and provide archives that are exactly identical to those in the active station. When a fault is detected in the Active unit, one of these inactive stations enters into active mode and automatically manages the plant by starting up the driver communication, recording engine and data collection functions to take over control where the primary server left off.
- The Inactive Server Drivers are kept on stand-by and do not communicate directly. They operate by receiving and sending variable value notifications to the Active Server and other servers. This is done in complete automatic and transparent mode. As a consequence a command can be invoked towards the field from any one of the Servers indifferently, while a change page operation, for instance, is performed locally as each unit processes its own graphics.
- The Inactive Server historicals (Historian, Data Logger, Recipes, Historical Log) do not operate directly in order to ensure the absolute sameness of recorded data. The appropriate system's redundancy functions perform in such a way so that data which is collected and recorded by the Active Server are archived in the same identical and transparent way on the other Servers as well. The synchronization mechanism always guarantees integrity and time precision of data.
- The Alarm state of the Active Server is redundant in all the other Servers. Any Ack. and Reset command executed on any one of the Servers will be transmitted to the Active Server which will perform the action.
- The Event list and Events on Variables of scripts are not executed in Inactive Servers.

## Supported Functions

The Fault Tolerance functions integrated in Movicon.NExT provide redundant management of the following functional modules:

- Communication Driver Manager
- Historian and Data Logger Manager
- Historial Log Manager
- Alarm Management

## Unsupported Functions

Recipes are currently not supported (at work).

Below is a list of the other unsupported functionalities whereby 'unsupported' means that the functionality is executed autonomously and individually by the relative server without automatic synchronization.

- Event Objects
- Alarm Dispatcher
- Schedulers

## Logic and Variables

A system variable is available on the Server that manages information such as local logic when the Redundancy Server is 'active' or 'inactive'.

## List of Servers

Even though the Movicon.NExT redundancy function can support up to 64 servers, **only 4 servers are currently guaranteed support.**

The names of the servers to be used in the projects must be inserted in the " names should be inserted by means of using the "Redundancy Server" settings. The order in which the servers are inserted on the list is very important as this defines their order of priority which will be used to identify the Active Server. During normal working conditions all the servers on the list are started up and operative in the plant as configured.

The Active Server will be the first one on the List and therefore in charge of managing driver communications and logging data on the Data Base according to the normal running of each project. The other Servers on the list are available and operative to perform the same functionalities as the Active Server but independently and according to their own configuration.

## Time Synchronization

It is important to consider the redundant Server system's time synchronization for greater congruity of historical data synchronization. Each PC is automatically synchronized with the Microsoft Windows main server clock according to the operating system's automatic functionality.

However it is also possible to synchronize the time of each Server with a Server Time that is different to that of windows for default. This can be done by changing the settings in the "Internet Time" controls of the Operating System's Settings.

### 4.11.3. Client Redundancy

The Movicon.NExT Client module, whether for local or remote server visualization, can always connect automatically to any one of the Servers started up in the redundancy system. This functionality is intrinsic to each Movicon.NExT Client and does not require any additional option to be activated on the license.

- Even though run as Client only, the project is defined with the Server List as previously described.

At the startup the Client checks the availability of a Local Server connection using the "net.pipe" transport if enabled in the project. If a Local Server is not available, the Client will check for a Remote Server connection using the network transport configured in the project (eg. "net.tcp").

The choice of Server to connect to is based on the Load Balancing which is revealed upon connecting to the server.

Therefore when connecting to a Server, each Client checks for a server with an 'optimal' workload situation. The Client will first try to connect to the machine's local Server if present by using the "net.pipe" transport when enabled in the project. The moment in which the Server becomes unavailable locally, the Client will try to connect to a remote server using a network transport such as "net.tcp" that must be enabled in the project. The choice of Server is based on the amount of workload detected when connecting.

When connecting to a Server, each Client must verify which server is the most appropriate to use at that moment before choosing the best one to connect to. This will ensure an effective distribution of workload in each server, above all in those situations where a lot of Clients are in operation, in order to obtain the best performances of the whole system at large.

After the Client has connect to a Server, the connection will be kept active until that Server is no longer available. In this case when the connected Server becomes unavailable (e.g. go into Fault), the Client will automatically try to activate an connection in on of the other redundant Servers available in the network.

The Automatic Connection and Load Balancing technology are both available in Movicon.NExT to ensure continuous Client Station service that is automatic and transparent to the user. At the same time this technology optimizes data traffic settings contributing to system efficiency and its performances.

#### 4.11.4. Redundancy Server

The Redundancy Settings can be configured by means of using the 'Redundancy' resource which is available in the "I/O Data Server" group settings. However, some of the Redundancy's advanced setting parameters are only available in the Properties Window in the specific 'Redundancy' group.

For further information please see the chapter headed: "Advanced Server Settings"



The Server's Redundancy functionality is based on the UDP Multicast protocol for which it is necessary to set the Gateway's IP address in the network form.

In cases in which the project is run on PCs without internet access, and therefore the use of the Gateway is irrelevant, you will need to set the Gateway's IP address in the network form with any IP address of the same subnetwork.

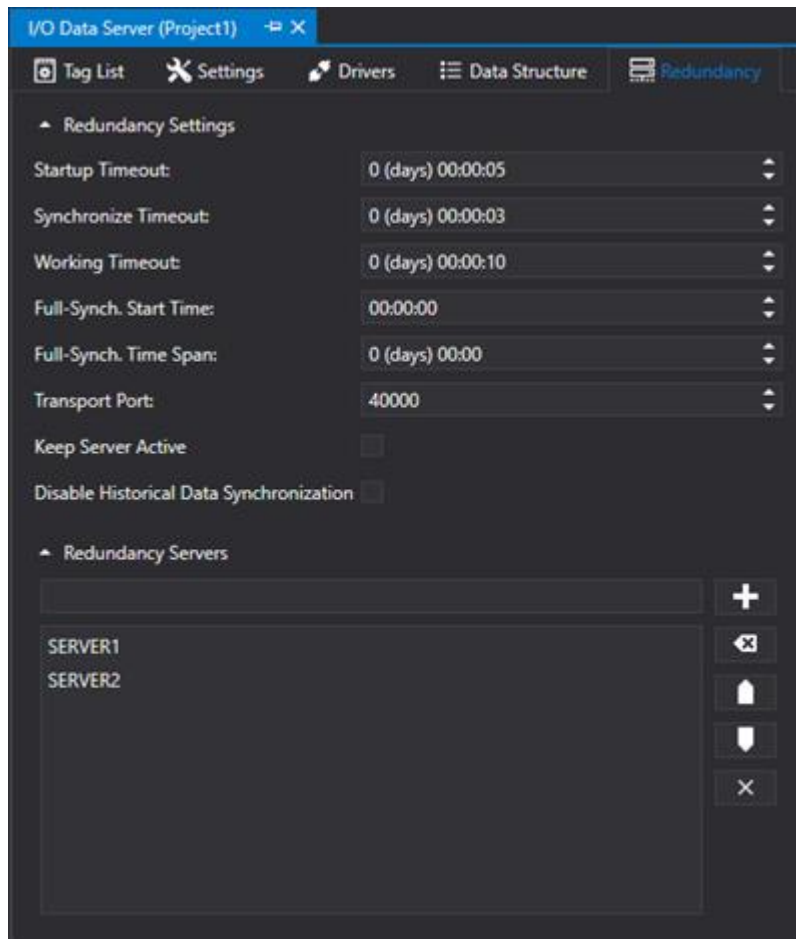


The Server's Redundancy functionality is only supported for PCs that belong to the same subnetwork (those with the IP address of the same subnet mask).

Therefore, It is not allowed to use this functionality if the PCs are connected to each other by means of a router.

The parameters available are:

## General Properties



### Redundancy Server List

This list of Servers participating in the redundancy are inserted in this box. Therefore the Host Names of the machines in question are to be inserted here.

Caution: You must insert the machine's Host Name and not its IP address. This is due to the fact that each machine often has more than one IP address and because the Redundancy manager always refers to the machine's name.



The order in which the servers are inserted also determines the priority of which Server will act as the system's Active Server. The first server on the top of the list of those started up and available will be the Active one which will have complete control of the system.



Attention! The hostname must be inserted in the Server list and NOT the IP address. In cases where the project is run on a PC in a network where a DNS Server is not available, you will need to set "<P\_Address> <PC\_Name>" aliases in the Windows 'Host' file to resolve the names of each PC involved in Redundancy.

### **Startup TimeOut**

Maximum time out used for searching for other servers in the network using the udp protocol. When the redundancy starts up it searches the network for those endpoints which satisfy the redundancy requirements and whose host name is on the list configured in the project currently running.

### **Synchronize TimeOut**

Maximum time used to wait for messages that the inactive server exchanges with the active server to maintain synchronization. These messages travel through the tcp channel and include the request for the initial tag and alarm values as well as the list of historical logs to keep synchronized.

In addition the same time is used as a ping interval that the inactive server uses to see whether the active user is still active.

### **Redundancy Timeout**

Timeout used for messages that the active server sends to other servers through the udp channel (for instance when a tag or alarm changes). This time is also used at the redundancy startup as the maximum timeout between the start and notification that the redundancy has started. If nothing initializes within this time an error is displayed.

### **Full Synchronization Start Time**

This is the startup time for the entire database. During the startup and shutdown phases of several drivers data misalignment may occur between the various servers. Full synchronization will realign the databases if this should occur. When the "Full Synchronization Time Span" is set with another value that is not zero, this property will be ignored and the "Full Synchronization Time Span" will be used only.

### **Full Synchronization Time Span:**

This setting is used for defining how often each full database synchronization is to be performed.

### **History Thread Pool**

The number of Threads to be used for handling the Historical synchronization. The "-1" value means that the same number of threads will be created to match the same number of existing CPUs or CPU Cores. Historical synchronization performance will increase when the number of threads is increased but this will effect the performance of other functionalities such as communications and animations for instance.

### **Max. Sync. Entities:**

This parameter involves the synchronization of the historicals and corresponds to the maximum number of records that are read by the active server for each synchronization job done by the other servers.

### **Transport Port**

Port used for transporting the redundancy functions.

### **Max Received Message Size (bytes)**

This is the maximum packet size that can be exchanged for synchronizing tags.

### **Max Retransmit Count**

Number of retransmissions of udp packets sent by the active server to send tag and alarm updates to the other servers.



The udp protocol does not control whether the packet has arrived at its destination to eventually re-send it if it hasn't. This parameter however allows the packet to be sent several times to reduce any eventual problems of recipients not receiving it; contrary to this by setting this parameter with a value other than zero will compromise the redundancies performances in sending information.

### Max Pending Transmit Message

Maximum number of pending messages. When the number of messages to be sent to Inactive Servers exceeds this limit, those exceeding this limit will be discarded. This is used only for updating tags in realtime. It concerns the maximum number of changes manage for each tag. If the number of changes to be sent exceeds the one send in this parameter, the oldest changes are removed leaving those most recent.

### Keep Active Latest Server Active

This option has effect **only when there are two servers on the list** and is used for keeping the server which has control active while the other server reenters into operation after a fault.

- When this option is enabled and there are only two Servers on the list, priority will not be given to the next active server on the list to take control. The control will remain with the server currently active when the other one returns into operation. For example, if a fault occurs and causes the first active server on the list to crash, the second server on the list will activate and take over full control. However when the first Server returns into operation, it will not take over full control until a fault occurs causing the second server to crash.
- When this option is disabled, or there are three or more servers on the list, this option will be ignored and priority will be given to the one according to their position on the list. When the highest listed Server returns active, it will take over full control.



In cases when enabling the 'Keep Server Active' option, and only two Servers are listed, a switch over to the active server can be done using the "RedundancySwitchActiveServer" command.

### Disable Historical Data Synchronization

This setting is used to deactivate the historical data synchronization (database) between the Servers that are on the redundancy list.

This function is handy when the databases reside on a machine in net instead of on each machine that runs the I/O Data Server and therefore centralized instead of distributed. In this condition, it is the job of the developer to ensure that historical data are always accessible (e.g. by using a redundant SQL Server database).

## Advanced Properties

### Max. No. of Synchro. Items

This parameter concerns the synchronization of historicals and corresponds to the maximum number of records that are read by the active for each synchronization job that the other servers do.

### Max. No. of Historical Threads

The number of threads to be used for managing historical synchronization. The "-1" value means that as many threads will be created as there are CPUs or CPU Cores. Increasing the number of threads will increase the historical data synchronization performances and compromise those of other functions such as communications, animations etc.

### Maximum Received Packet Size

Maximum packet size that can be exchange for synchronizing tags.

### Max. No. of Send/transmission retries

Number of udp packet transmission retries sent by the active server to send tag and alarm updates of the other servers.

The udp protocol does not check whether the packet has arrived at its destination to re-send it if it hasn't. This parameter allows the same packet to be send several times to reduce the probability that it does not get received by one of the recipients. However, setting this value other than zero however compromise the redundancy performances when sending the packet.

### Redundancy Max. Pending TransmitMessages

The maximum number of pending messages. When the quantity of messaged to be sent to Inactive Servers exceeds this limit, those in excess will be lost. This parameter is only used for real-time tag updating. It concerns the maximum changes manages for individual tags. If a number of changes for one tag exceeds the value set in this parameter, the oldest changed will be removed leaving the most recent ones.

### Transport Security Mode

Security level to be used for when transporting data. The options are:

- None
- Transport
- Message
- TransportWithMessageCredential

### UDP Service IP Address

The "Listening" Area for UDP packets. The 224.0.0.1 value indicated the group consisting of all the LAN network Hosts.

### UDP Client IP Address

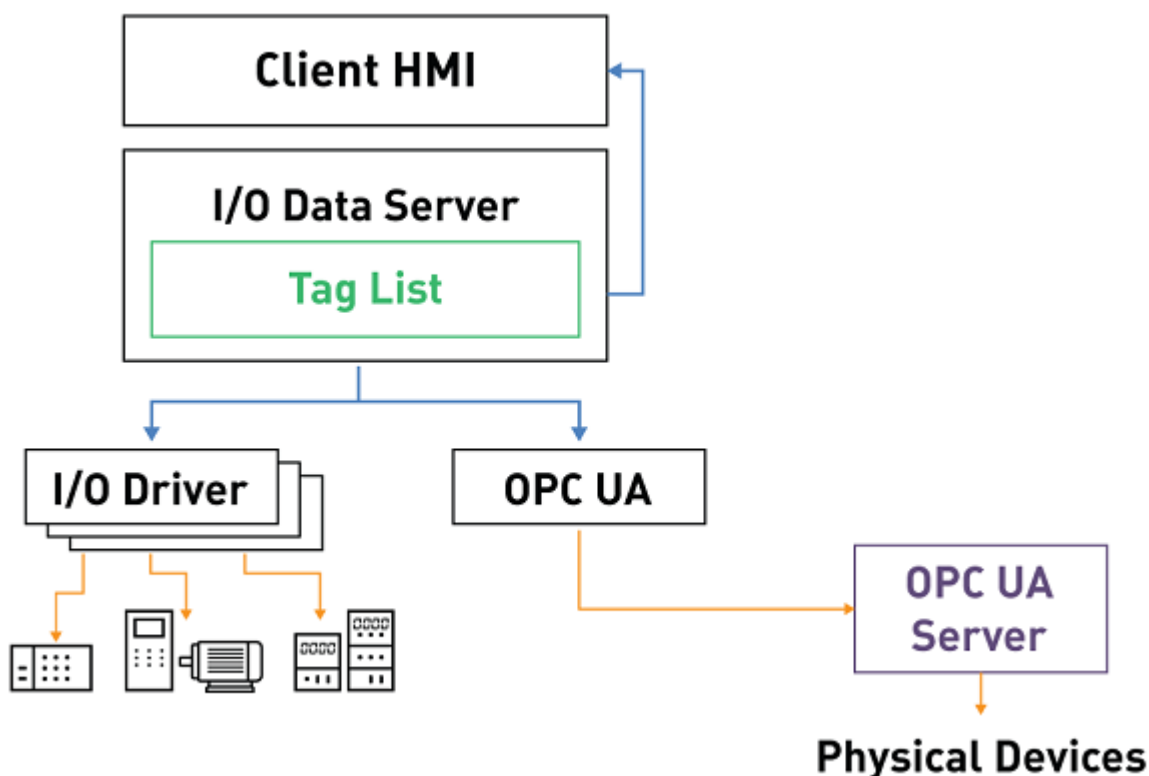
Defines the area for sending UDP packets. The 224.0.0.1 indicates the group consisting of all the LAN network Hosts.

## 4.12. Communication Drivers

### 4.12.1. General Driver Concepts



One of the main tasks of the I/O Data Server module is to manage communications with the field devices using different communication channels that manage protocols towards the physical I/Os of external peripherals.

The platform is therefore capable of communicating in real-time with field devices using communication drivers that are included with the product or technology that connects to third party OPC UA Servers. Whatever mode is used all the information collected from the field physical addresses is associated to Tags which are found in the Tag List resource used for the Server's Address Space. The Tag List renders all real-time information visible to all the resources and modules of the entire Automation Movicon.NExT platform whether Server or Client.



This diagram shows an example of an architecture using I/O Drivers or the OPC UA technology to communicate.

The Driver configuration specifications are described in the relating sections of this documentation according to each driver's specific communication protocol type (e.g. Siemens S7. Ethernet-IP, Modbus, Omron and others) along with the OPC UA standard communication technology.

-  The I/O Data Server Communication Drivers are installed with the same Server. However, additional drivers, or others created by third parties, may be become available after the installation.
-  Movicon.NExT has been designed to enable expert users to create their own specific communication drivers and integrate them within the platform. However, a SDK is available for developing custom protocols and support and advice is also provided if needed.

## Drivers

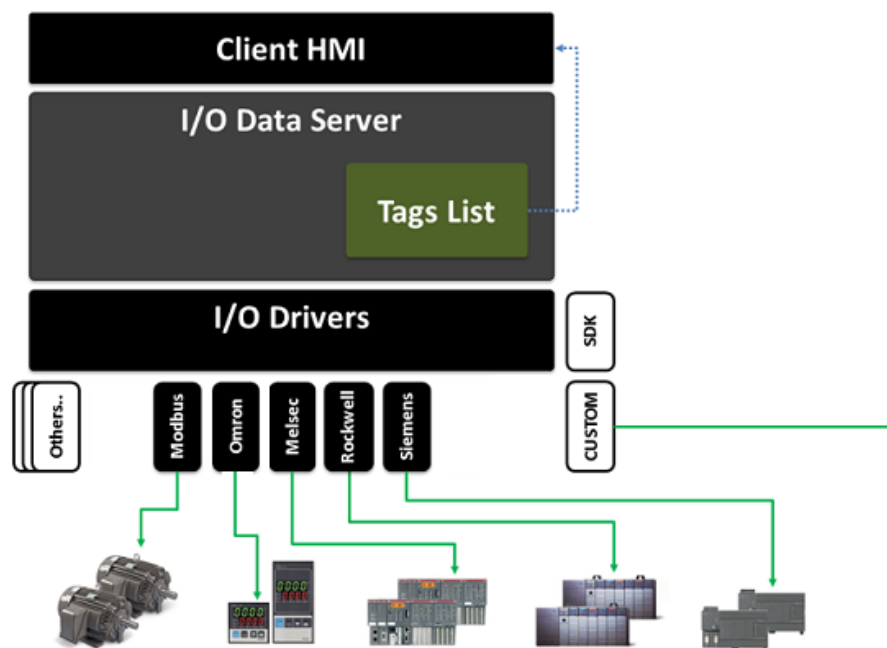
The Communication Drivers are components of the I/O Data Server which allows you to manage specific communication protocols to consent the physical connection to field devices.

They are built form dynamic ".NET" libraries (.DLL file) which, according to exception-based logic, notify information received from device's memory areas connected to the I/O Data Server memory areas and

vice-versa, according to the predefined settings for the specific communication protocols. The I/O Data Server, which is the gathering point of dynamic information, places this dynamic information in Tags by means of the address space, to the disposal of all platform modules.

Communication Drivers are available in the I/O Data Server and the list is in continuous expansion. The most commonly used drivers are those for PLC devices (Siemens, Schneider, Rockwell, Omron, Mitsubishi, Saia, Panasonic, etc.), fieldbuses (Profibus, Profinet, Ethernet/IP, EtherCAT, Powerlink, etc.), RTUs and instrumentation (Modbus, etc.) or for infrastructure networks (IEC61870, IEC60850) and civil systems (Konnex, Bacnet, Lon, etc.).

The system's open and modular architecture consents to integration of third party Drivers in the I/O Data Server, to ensure openness needed to the integration of new communication protocols.



*Generic connection diagram between Movicon.NExT and field devices. The list of available drivers is long and constantly being added to.*

#### 4.12.2. Communication Drivers

One or more communication drivers can be inserted and configured as needed within the Movicon.NExT project's I/O Data Server. Each communication driver inserted in the project should be configured with specific characteristics depending on the device's propriety communication protocol. In order to do this, you will need to refer to the guide of each specific driver for details on addressable memory areas and communication parameters.

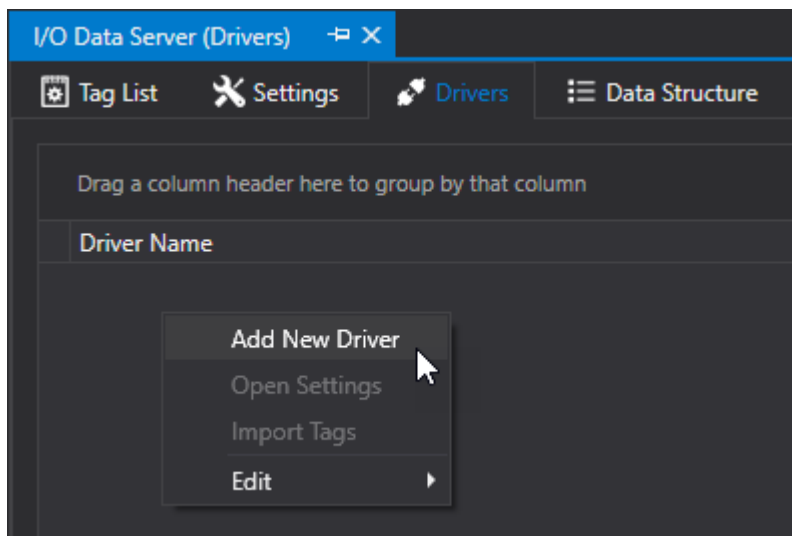
In this topic we will look at the general concepts on using a I/O Data Server driver which are common to all the other drivers available in the platform.

### Adding a new Communication Driver

In order to insert a new communication driver, you will need to open the I/O Data Server resource and select the "Drivers" pane.

Right click to select the "Add New Driver"(fig.1) option, or use the project menu to select the "I/O Data Server" item and then the "I/O driver" to add the desired driver (screenshot 2).

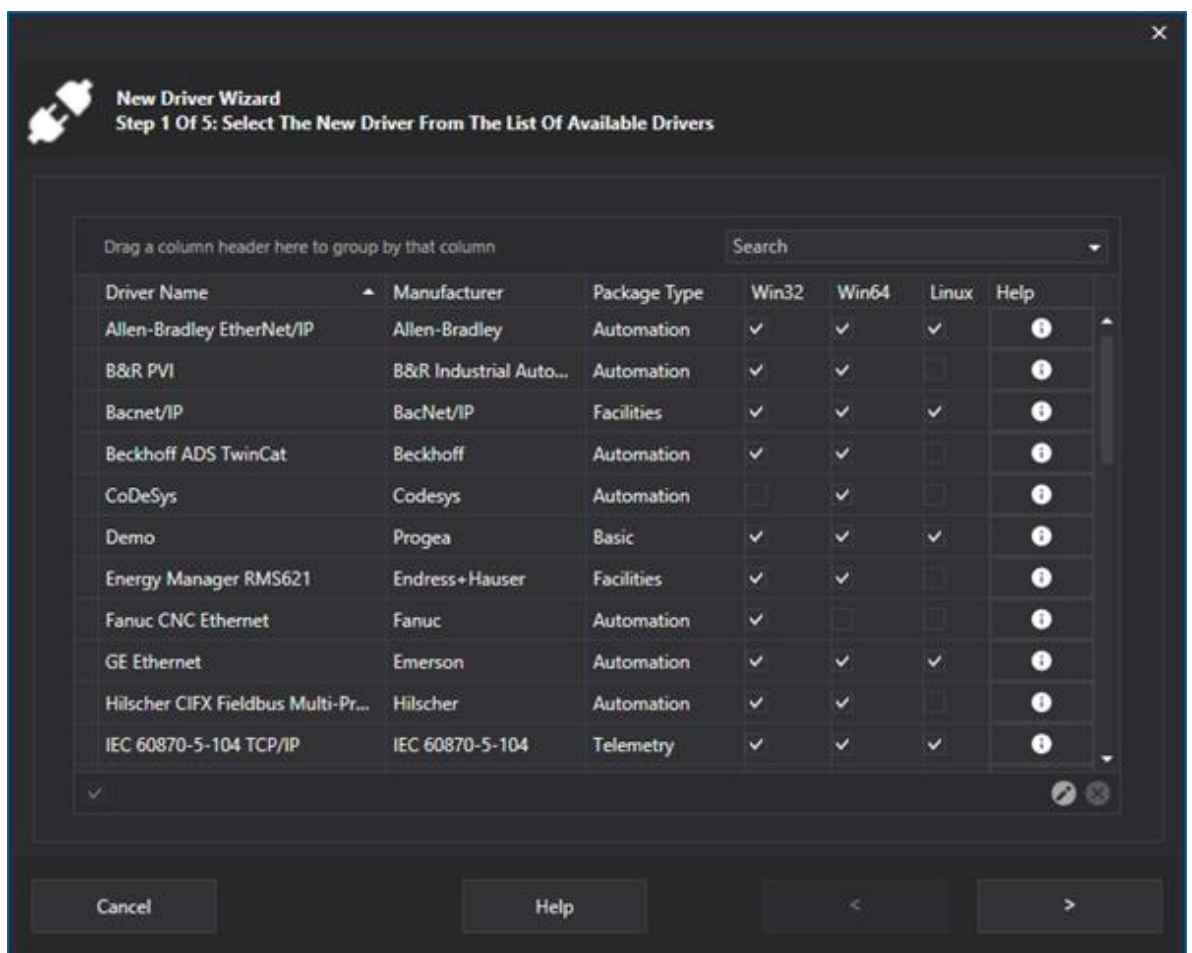
Added drivers will be listed in the Driver pane where it is also possible to change the driver's settings or remove it from the project.



Screenshot 1

When executing the command to insert the new Driver in the project, the Wizard will show to help you configure it in the most simplest way possible. The first thing to do is select the driver you wish to install from the list of drivers available in the system.

The list of drivers shows which drivers are available grouped by the name of the Manufacturer and by device or protocol type.



*Wizard window used to select the Driver to install in the project.*

Select the desired driver can continue on using the configuration wizard as described below.



Attention! Drivers available for Movicon NExT also require their basic library .dll files which must be in the same version (or later) of the driver you wish to download.

Attention! All the Movicon NExT drivers that are available for download are compatible with the program 3.1.109 and later versions.

It is advised to also download the Drivers.xml file containing the list of updated Drivers in those cases where the selected driver is not displayed in the list of those available in the program.

## Driver configuration wizard

The Driver configuration wizard has three steps/forms: "General Settings", "Channels Settings" and "Stations Settings".

Some of the parameters presented in the configuration phase are common to all Drivers, while others are specific to each Driver and depend directly on the protocol type used by that Driver.

### General Settings

The "General Settings" allows you to define some of the Driver's general settings.

For further information concerning the general properties of the various drivers, please refer to each listed Driver's specifications.

### Channel Settings

The "Channels Setting" form allows you to define the Channel which the Driver will use for communications.

The parameters presented here are those that allow you to manage communication velocity (such as the Polling Time) and communication type.

The parameters presented for serial type Drivers, for example, include those for the COM port settings such as the Baudrate, while those presented for Ethernet type drivers are those which include Hostname (or IP address), Port and so forth.

### Station Settings

The "Stations Setting" form allows you to define the Station which the Driver will communicate with through the Communication Channel.

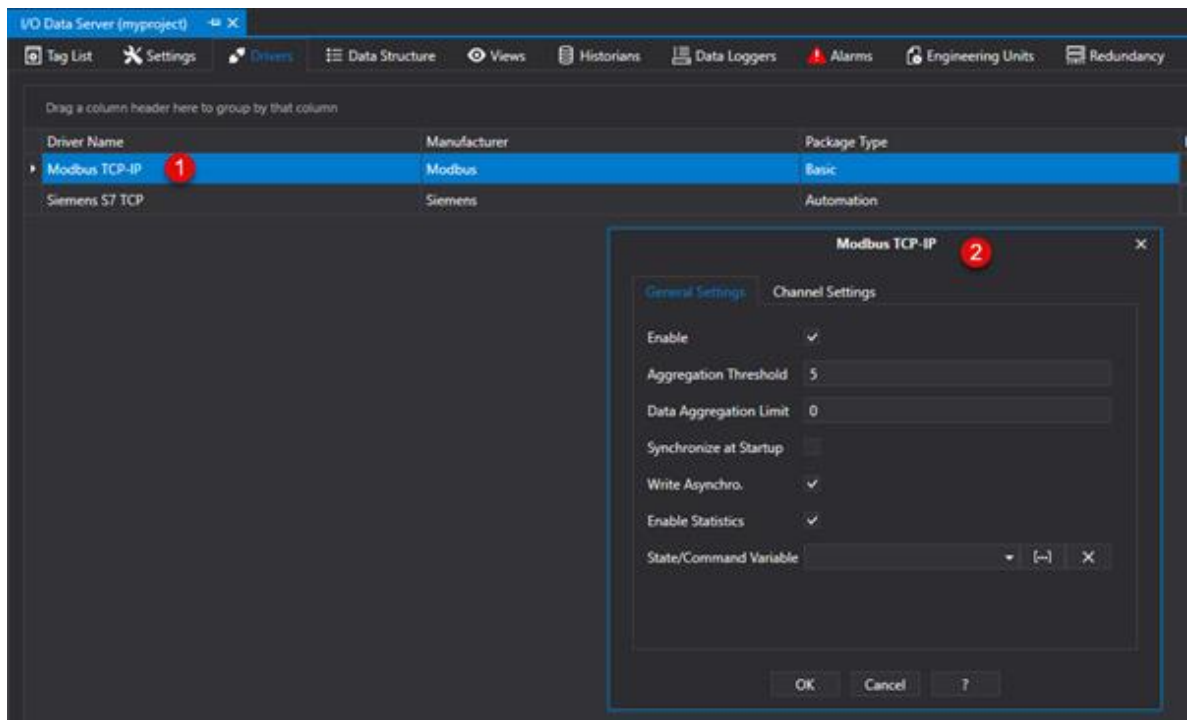
The parameters of a Station settings may include, for example, the slot with the CPU is mounted in the rack, or the PLC model.

## Modifying Driver Configurations

The Driver configure wizard allows you to define one Channel only to which one Station only can be associated to allow one Driver only for communicating with one device only.

If you need to communicate with several devices using the same driver, you can add more communication Channels and stations.

By double clicking on the driver's name, a pop-up window will appear allowing you to modify the Driver's settings:



By using this window, you will be able to change the General Settings, or access the Communication Channel Settings to add other Channels or modify those already existing.

## 4.13. Server (OPC UA Browser) Diagnostics Window

When starting up the Server side from the development environment manually, using the "Start Server" command, a Server OPC UA Browser window will automatically open showing various information. The information shown in this window includes Server diagnostics, list of set alarms, list of Tags that the Server has published. In addition, by using the window's 'Live Data' section you will be able to view real Time Tag values that can be modified if Tag is writable.

The Server's diagnostics window is in fact an "OPC UA Browser" window that can also be used to connect to an OPC UA Server to retrieve information.





# 5. Client Configuration

## 5.1. Client connections to Server

The Movicon.NExT platform uses a Client-Server architecture with the standard OPC UA-based data information model.

This model ensures maximum project client and server configuration flexibility.

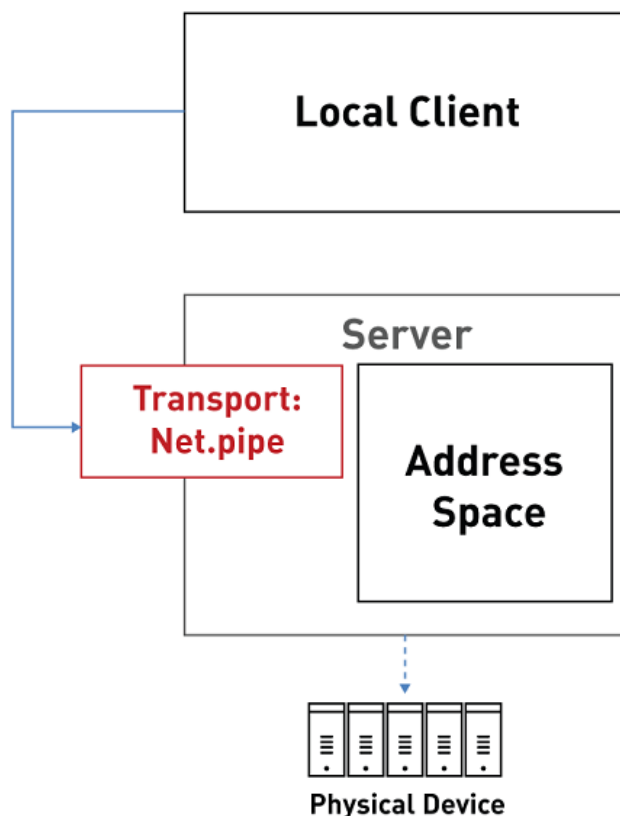


Warning! Client connection to Server needs to be enabled on the License on the Server. Except for special licenses, the Networking function is usually enabled and consents Movicon.NExT Client access to a Movicon.NExT Server. Third party OPC UA Client access requires the activation of the OPC UA Server option on the purchased license.

### Client Connection to Local Server

The Client connection to a Local Server is always consented and provided for default and only needs a Client-Server license to enable this.

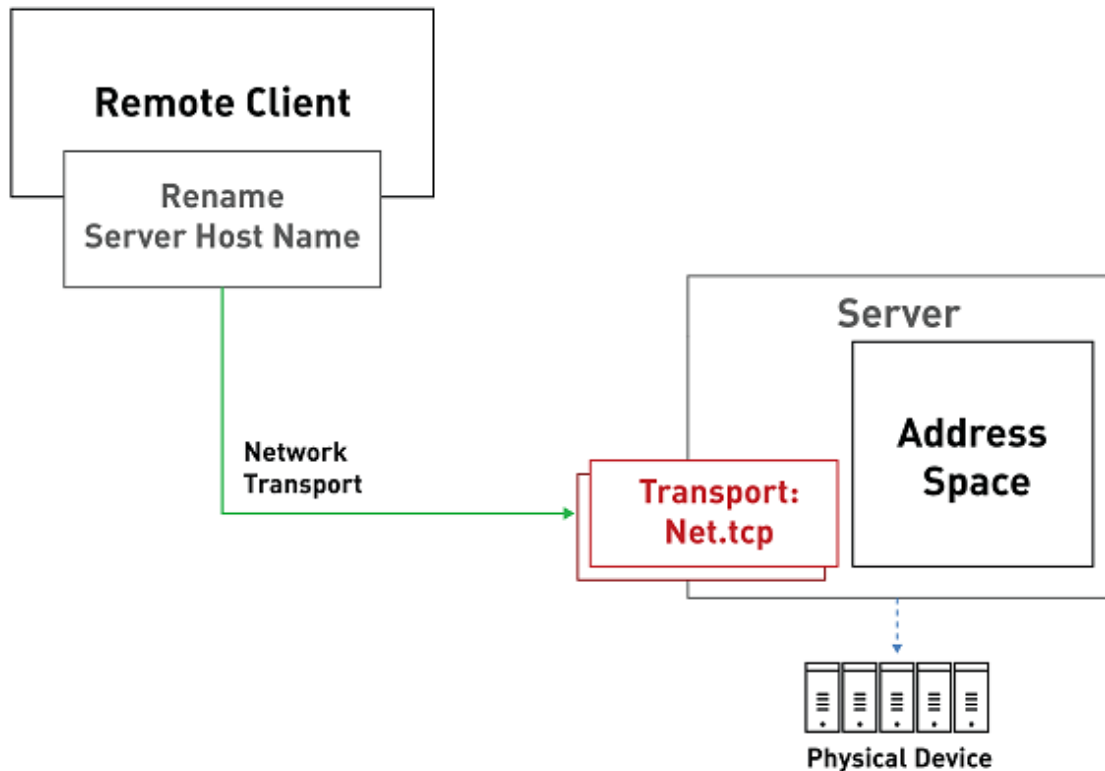
The use of a local transport (e.g. "net.pipe") or a net transport (e.g. "net.tcp") does not have any particular influence on this type of architecture.



### Client Connection to a Remote Server

The first thing to do, in addition to enabling the remote Client connections option on the Server, will be to define a net transport in the Server's "General Settings" (e.g. "net.tcp").

The choice of net transport to use is entirely based on your security needs or any other specific need you may have as indicated in the paragraph on "Data Transport Definition"



While creating the project, the wizard will propose three Client-Server architectures. Therefore, it may be possible that the net transport that you need be already configured during this phase.

The second thing to define is the Client connection settings.

In the connection parameters, you will need to specify the name of the Server, which the Client is to connect to, in the "Hostname Server" property as shown below.

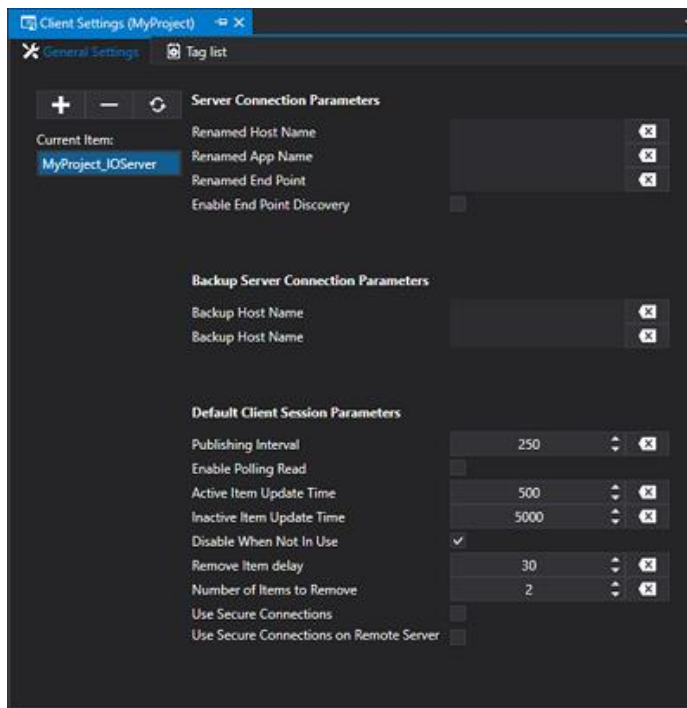


You should also select the "Enable End point Discovery" option as well.

## 5.2. Client Settings

This settings window can be accessed through the "Client Configuration > Client Settings" resource from the project tree and allows you to define the Client's connection parameters towards the Server.

It is also possible to redefine connection parameters such as the name of the Server to connect to or the name of the Application if renamed.



## Server Connection Parameters

### Renamed Host Name

This parameter consents the name of the Server, which the Client is to connect to, to be entered here. It will be used when the Server is remote, meaning that when the Client and Server are running on different machines. At runtime, the client's connection endpoint will be automatically be redefined by substituting the "localhost" with the name of the Server indicated in this parameter.



The substitution of the Server name will only be done correctly if the project's default endpoint has been left set with the Server name of "localhost".



IP address usage is not supported.  
You will need to specify the server's hostname.  
If a DNS server is not present in net, you can modify the Windows "hosts" file.

### Renamed App Name

This parameter is used when changing the Project's Application Name after having already assigned Tags. In the condition, the new Application Name is entered here, while the name of the name of the set of parameters being edited should have the old Application Name. In this way, at runtime, the connection endpoint to the Server will be redefined by substituting the old Application Name with the new one. Please refer to information on "Renamed App Name" in this page

### Renamed End point

This field is used to define the whole endpoint that the Client should use to connect to the Server. In this case, the "Server Hostname Server" and "Application Name" parameters will no longer be necessary and ignored. An example of an endpoint to enter would be:

```
net.tcp://ServerName:62846/Project1_IOServer
net.tcp://192.168.0.49:62846/Project1_IOServer
```

### Enable End point Discovery

Enabling this option will enable the Client to connect to the Server using any one of the transports published by the Server in cases where the transport defined in the default endpoint is no longer available. Supposing for example that while in project design mode, only the "net.pipe" transport is being used but when in runtime the Server only exposes the "net.tcp" transport, the Client will not be able to connect to the Server. By enabling the "Enable Endpoint Search" option, the client will then be able to use the "net.tcp" transport to connect to the Server.

## Backup Server Connection Parameters

### Backup Host name

This parameter is used to enter the name of the Backup Server to which the Client is to connect to in cases where the main Server Host name cannot be reached. It is used when the Server is remote in a situation where Client and Server are running on different machines.



The substitution of the Server name will only be performed correctly when the project's default endpoint has been left with "localhost" as Server's name.



IP address usage is not supported.  
You will need to specify the server's hostname.  
If a DNS server is not present in net, you can modify the Windows "hosts" file.

### Backup Application Name

This parameter is used to enter the name of the Backup Application to which the Client is to connect to in cases when the main Server Hostname is unreachable.

## Default Client Session Parameters

### Publishing Interval

This value, which the Client passes to the Server when the Session is opened, determines the time with which the Server sends update notifications of tag subscribed by the Client.

### Enable Polling Read

This property forces the Driver to read Tags with polling Tasks instead of using OPC UA Subscriptions.

### In Use Tags Update Time

This value determines the time with which the Server keeps Active Tags updated through the communication driver by setting it as the minimum reading task execution time.

Active Tag are those that the Client has subscribed to the Server and that are currently in use on the Client in the following situations:

- In use in visible Screens.
- In use in invisible Screens kept in memory (Keep in Memory option).
- In use in invisible Screens pre-loaded at startup (Pre-loaded Screen List option)

### Not in Use Tags Update Time

This value determines the time with which the Server keeps Inactive Tags updated through the communication driver by setting it as the minimum reading task execution time.

Inactive Tag are those that that the Client has subscribed to the Server but are not currently in use in any visible Screen or in memory.

#### **Disable When Not in Use**

When enabling this property, the "Item Not Active" state is set when the Tag is not in use in the Client.

#### **Remove Item Delay**

This parameter defines the delay time that when expires will remove the subscribed Inactive OPC UA Items according the number of items set in the "Number of Items to Remove" parameter. Time is expressed in seconds.

#### **Number of Items to Remove**

This is used to set the number of inactive OPC items to remove at each time interview defined in the "Remove Item Delay".

#### **Use Secure Connections**

Enabling this option, the Client will connect to Server connections declared safe by the appropriate certificates whether they be local or remote. However, in cases where the Server does not expose safe connections, it will expose the next available connection even if not safe.

#### **Use Secure Connections on Remote Server**

Enabling this option, the Client will connect only to Server connections declared safe when the Server does not belong to the same local Client network. However, in cases where the Server does not expose safe connections, it will expose the next available connection even if not safe.

## **Using Child Projects in distributed Client-Server Architecture**

The Child and Parent technology is available in Movicon.NExT Client-Server project network architectures. This technology allows you to create projects distributed in a network that automatically connect to a referenced Server.

Please consult the chapter on Movicon.NExT Child Projects for further information.

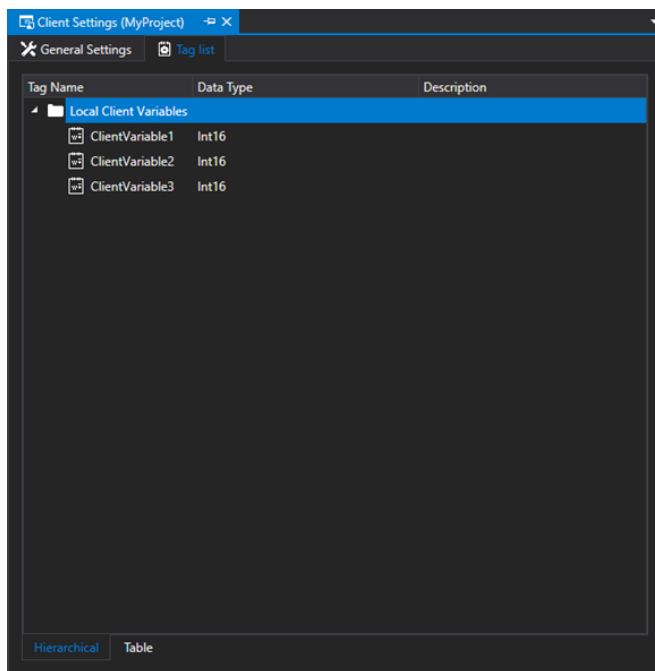
## **5.3. Client TAG list**

The local project tags are tags that the programmer can create within the project and use in screen controls or script. The Local Tags are not deployed in the Server's address space and are only visible at Client level.

The Local Tags, such as Server Tags, can be created with names defined by the programmer and of different types grouped in folders at several nested levels. However, local tags have certain limits in respect to the Server Tags such as:

- They can be defined with a "Data Type" but not as "Model Type"
- They cannot be exchanged with the field using a communication driver
- They cannot be assigned with engineering units
- They cannot be retentive
- They cannot be associated with alarms, historical or Data Logger
- They cannot be managed with user access levels
- They cannot be Structure, Digital or Enum type but only simple or Array types

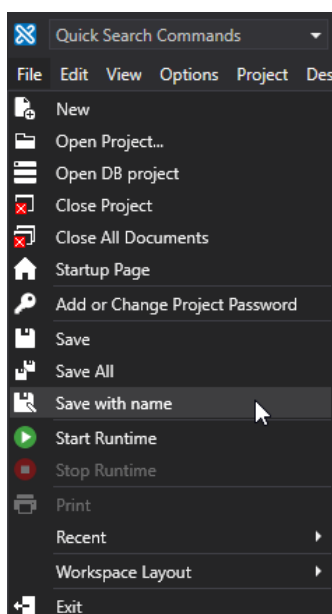
The Local Variables can be created using an appropriate editor that can be accessed from the "Local Variables" tab once opening the "I/O Data Server" resource. They can also be created directly from the Tag Browser window which is also used for assigning variables to objects.



It is possible to identify whether the variable associated from a control is Local or Server type by the notation shown within the brackets after the tag's name in the title of the object's "General Properties" window or in the tooltip which appears when clicking on the first small button at the top of the object's menu. The notation within the brackets will show "TemporaryVariable" when the variable is a local type or the Server's AppName, for example "Project1\_IOServer", when a Server Tag.

## 5.4. Changing Application Names

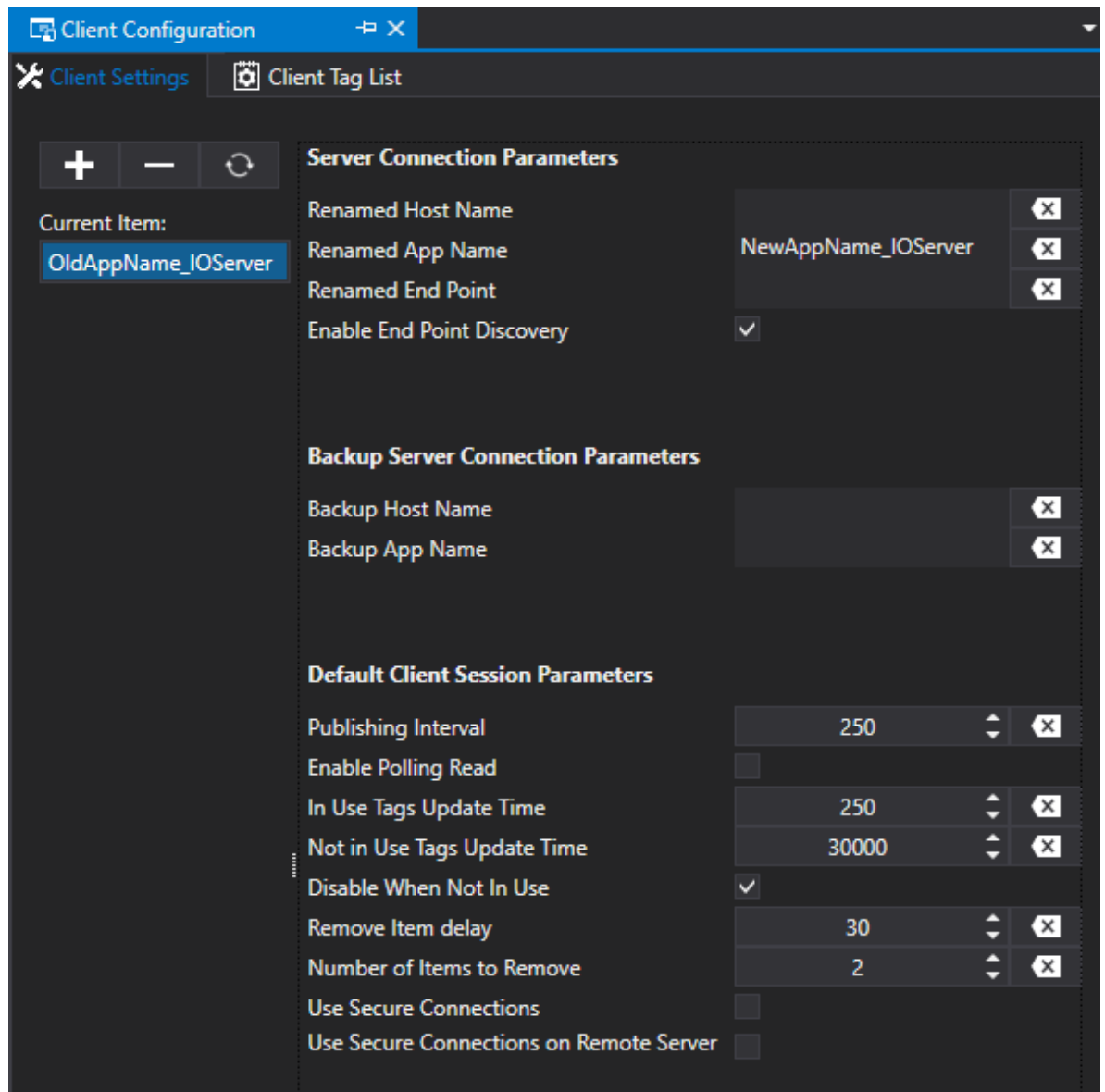
The "Save with Name" command from the "File" menu can be used to select a new name for the project as well as the target folder.



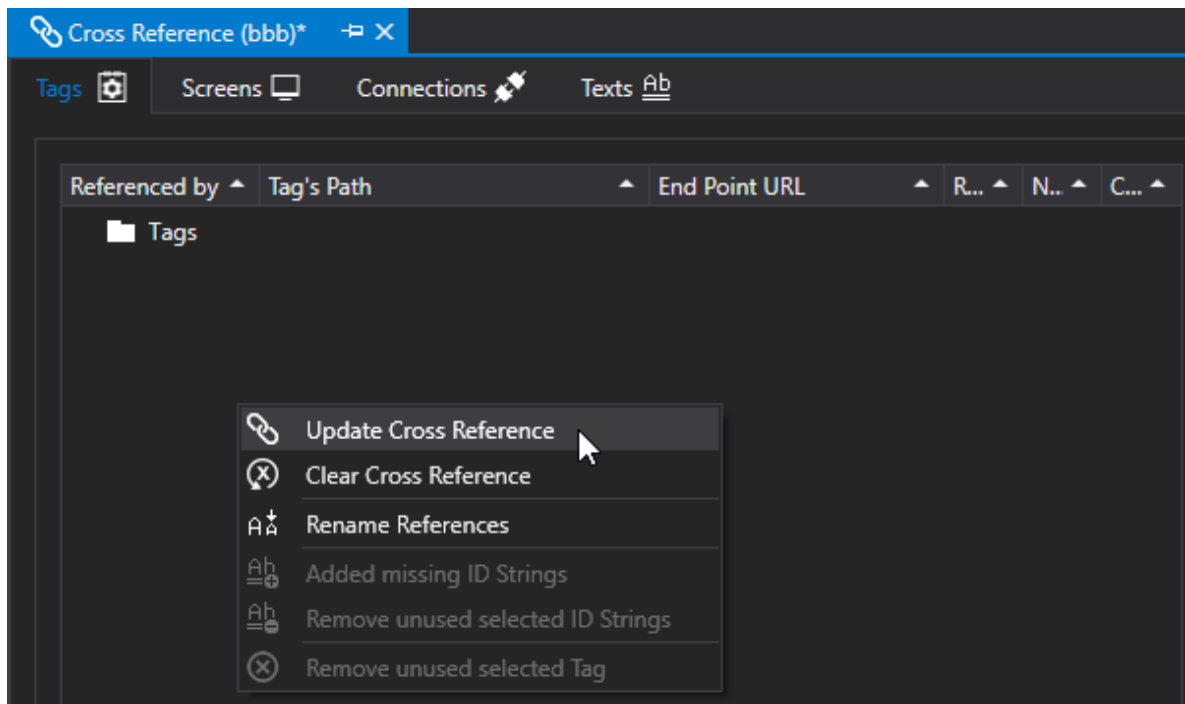
If you also wish to change the name of the application in addition to the name of the project, you will need to redirect the resources previously connected to the application's old name.

There are two ways in which to reconnect tags connected to the objects:

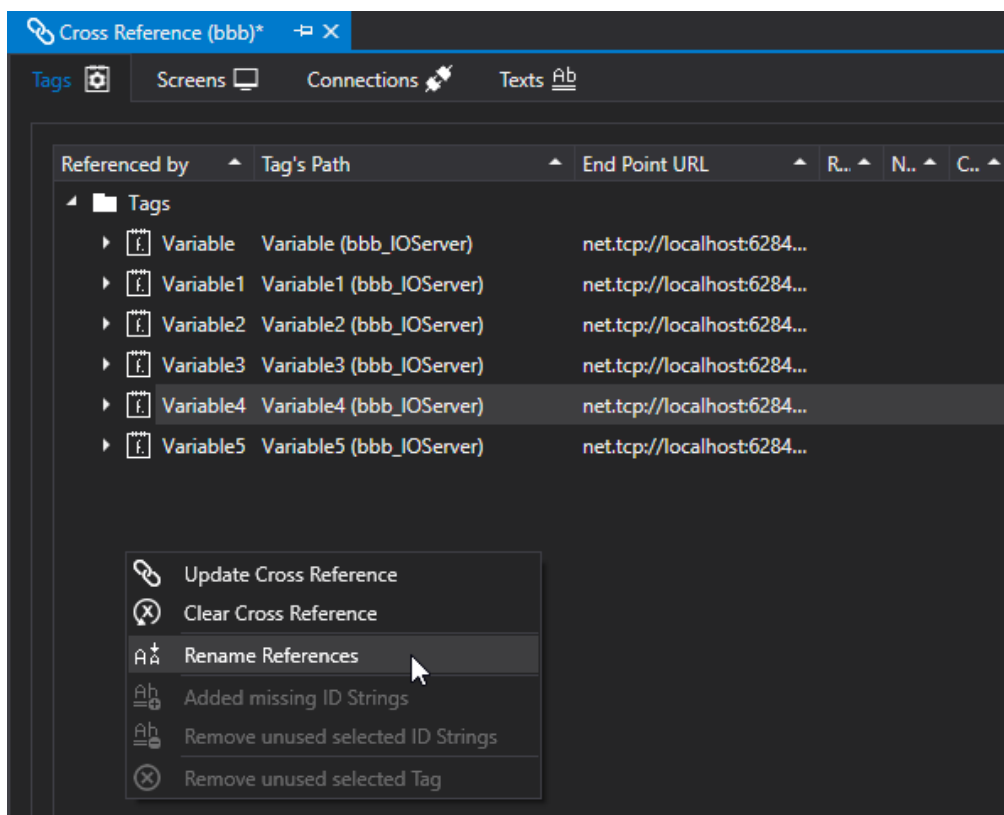
- Redirecting the application's old name (used in the resources) to the application's new name:
  1. Open the "Client Configuration" dialog window from the main project tree.
  2. Click the "+" button to create a new Application name.
  3. Enter the application's old name in the "Current Item" field.
  4. Enter the application's new name in the "Renamed App Name" field.
  5. Mark the "Enable End Point Discovery" option to activate it.



- Using the "Cross Reference" tool to redirect the objects directly in the resources towards the application's new name:
  1. Save the project.
  2. Go to the project tree and open the "Cross Reference" tool.
  3. Execute the "Update Cross Reference" command with a right click or select the command from the "Cross Reference" menu.



4. Select the resources to analyze and press ok. The Cross Reference will close all the project's resources and list the contents and detected connections in the appropriate tab.
5. Right click to execute the "Rename References" command for the tags or select the command from the "Cross Reference" menu to update the tag connections with the application's new name.





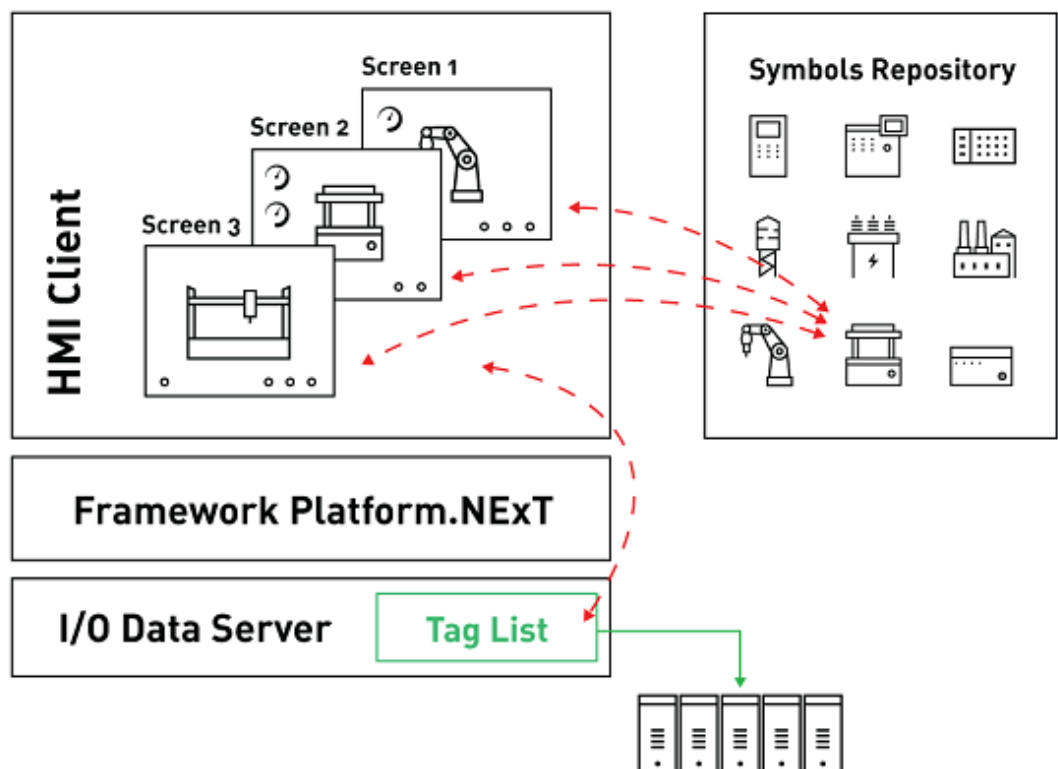
# 6. Graphics

## 6.1. Screens

### 6.1.1. Introduction to Screens

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

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



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

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

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



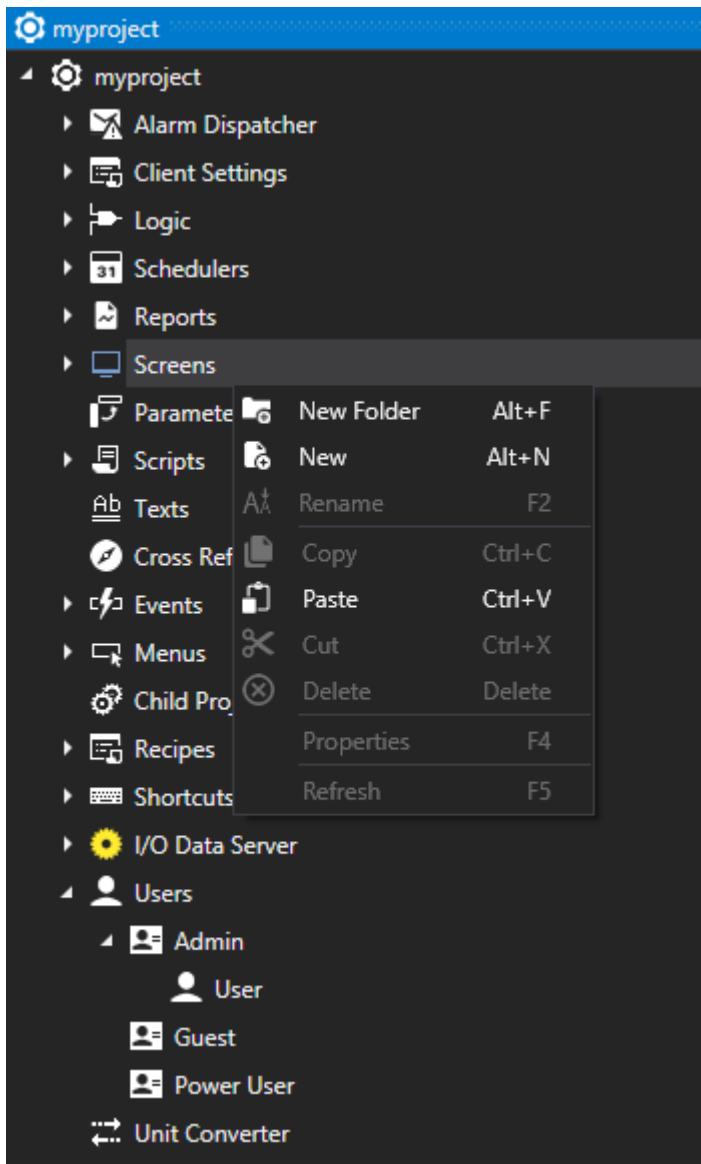
The Screens are run as "Clients" of a project. Their graphical content is connected to Server variables, being both the platform's data Server and third party OPC UA data server.

6.1.2. Inserting new Screens

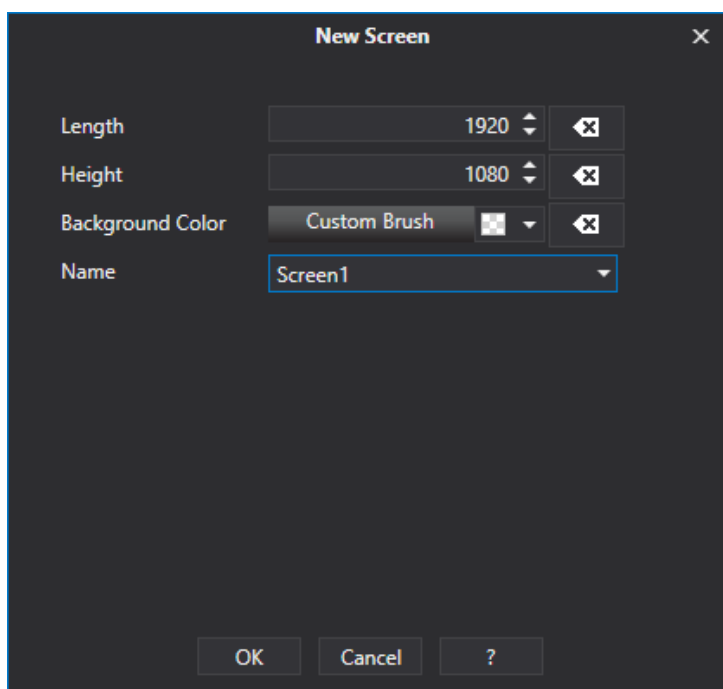
There are two different options to choose from when needing to insert a new screen in the project. You can either add a new screen or create a screen "Template".

New Screen

A new screen can be added to the project by right clicking on the project's 'screen' icon or by using the "Alt+N" shortcut.



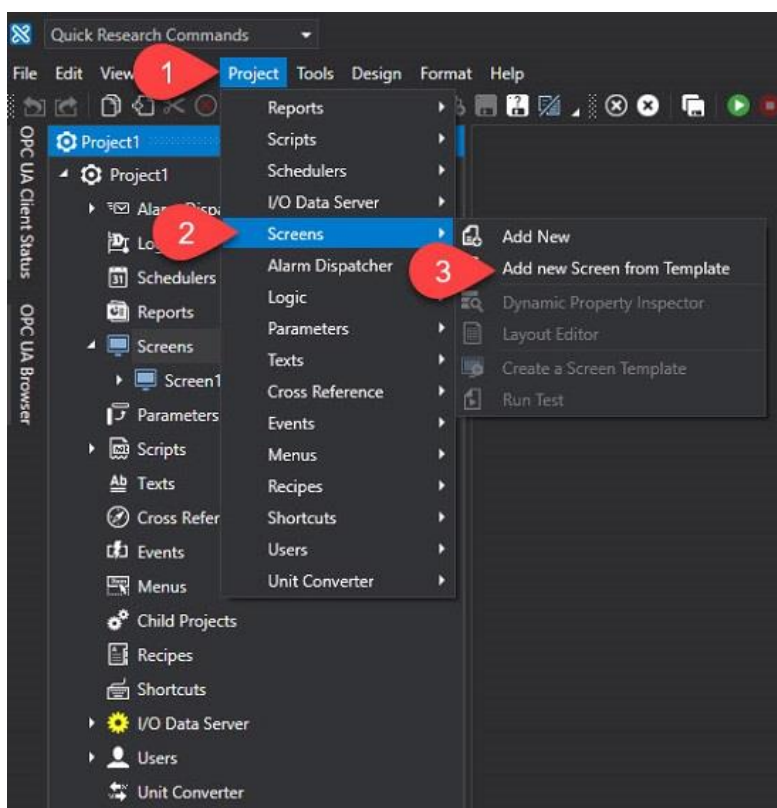
The below popup window will show allowing you to choose the Length, Height, Background Color and Name.



These parameters you select will be retentive and will be saved as standard setting for subsequent screens created within the project.

### Screen Templates

By using the "Add new Screen from Template" command, a window will open through which you can select the Screen Template to insert the Movicon NExT project.



This option offers a selection of ready-to-use screen models with different formats for different type of uses. These model types vary from simple 'empty' screens to those ready with the most commonly used graphical elements.

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

When creating a new screen, you will need to select one of the Screen Template and then confirm with "Ok".

Once confirming the new Screen will be created in the project according to the model you have chosen. You can then change the screen's properties by means of using the Property Window.

## Creating new custom Screen Templates

Screen Templates are screen models that are used for creating new project screens. Each time a new screen is inserted a Screen Template can be selected to use its characteristics for creating the new screen. The Screen Templates can also be created by the programmer to realize their own library of screen models to use as required.

To create a Screen Template simply select one that has already been configured in the workspace or configure it as pleased by inserting the desired symbols and objects. At this point, it is necessary to "save" this Synoptic as a "Template", using the command Project-> Synoptics -> "Create a Synoptic Model".

Once this has been completed the screen should appear among those Templates available when creating the next new screen.

The Template Screens are saved in the following folder:

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



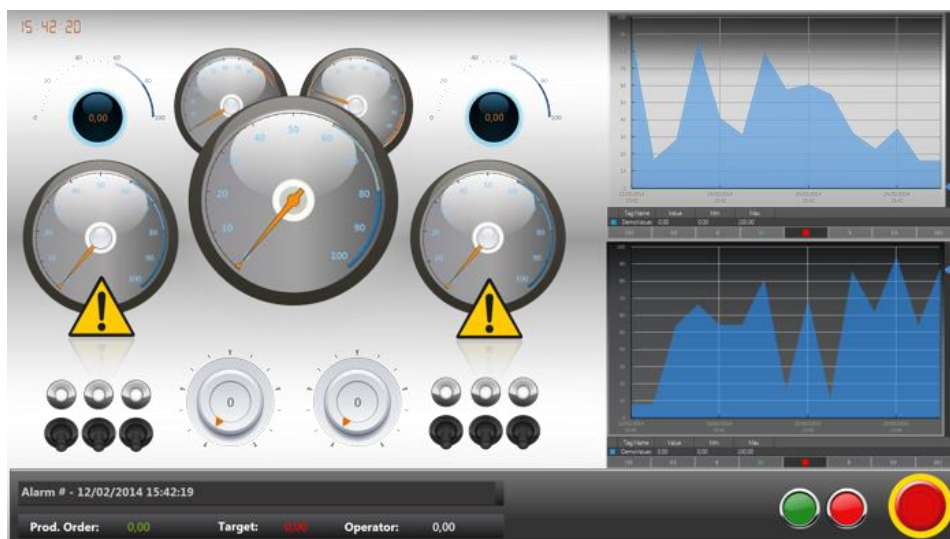
Attention: Screen Templates are deleted or removed by using the files in this folder.



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



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



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

### 6.1.3. Screen Properties

The Screens are project graphical pages in which various objects, symbols and controls can be inserted for managing the HMI graphical interface. The Screens are managed by the project Client side. Each screen has two property group settings. The first group is used for defining the screens aspect within the 'Tile Page' and its geo-localization co-ordinates. The second property group is used for customizing the screen's functional role when opened in runtime.



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

## Style

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

### Fit In Window

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

### Resize Screen Controls

When enabling this property, the screen controls will resize without applying the zoom. This property only appears when enabling the 'Fit in Window' property.



Enabling this property will lose support to the following animations:

- Move X;
- Move Y;
- Height;
- Length;
- Scale;

- Composed Movement.

### Background Color

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

### Window Style

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

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

### Window State

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

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

### Tile Size

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



Editable when selecting the screen from the project tree structure.

This property can be set with the following values:

- ExtraSmall
- Small
- Large
- ExtraLarge
- live

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

### Tile BackColor

When a screen is visible in the Tile Page, the background color of the rectangle, which identifies the screen, will assume a color at random. This Color property is used to specify a background color for the square that identifies the screen in the Tile Page.

When this property is left with the default value, the background color of the square will change to a different color at each startup.



Editable when selecting the screen from the project tree structure.

### Tile Description

This property is used for associating a description to the screen. This description will be displayed within the square representing the screen on the Tile Page. In this case, an automatic scroll between the screen's name, preview image and description will be applied within the square.



Editable when selecting the screen from the project tree structure.

### Advanced: Show in Taskbar

Allows the screen button to be shown on the Taskbar.

### Advanced: Resize Mode

This property is used for defining the screen's resizing mode. The options are:

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

### Advanced: Window Startup Position

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

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

### Advanced: Hide Scroll-bars

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

### Advanced: Show Navigation Buttons

Shows the navigation buttons at the side of the window.

### Advanced: Show Header

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

### Advanced: Show Tab Strip

Shows a navigation bar indicating the last screens displayed.

### Advanced: Show Project Toolbar

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

## Layout Properties

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

### Width

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

### Height

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

### Advanced: Top Position

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

### Advanced: Left Position

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

### Advanced: Zoom Level Cache Mode X

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

### Advanced: Zoom Level Cache Mode Y

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

### Advanced: Layout View

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

### Advanced: Layout View Editable

Allows the screen layout to be edited in runtime.

## General Screen Properties

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

### Visible

When setting this property to True, the screen will be made visible in the Tile Page when the project is started up in Runtime. The screen will be represented with a square containing the name of the screen



and a preview of its image. The screen will open by simply clicking this square. When this property is set to False, the screen will not be available in the Tile Page.

### Window Opacity

This parameter is used to define the screen's transparency level. This parameter is taken into consideration by the screen only when opened in Synchro mode (Modal or Frame, the "Synchro Popup" option must be disabled), or when the screen is opened outside the runtime shell. This means when the screen is opened in a different monitor to the one set for default. Otherwise this parameter will have no effect.

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



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

### Hide Layout Screens

This Layout Screens to be hidden when the Screen is open. In this way, each Layout Screen can be set to show or not. For further information on Layout Screens, please see this topic .

### Advanced: Window Transparency Only Inactive

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



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

### Advanced: Scan Children Elements

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

### Advanced: Disable Zoom

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

Screens can be zoomed at runtime When leaving this option unchecked. Screens can be zoomed by using the mouse wheel while keeping the CTRL pressed or by using the "pitch and zoom" in Touch Screens. To bring the screen back to its original size in Touch Screens, simply double tap it.

When zooming the screen so that it becomes bigger than the screen window, the Scroll bars will show if the "HideScrollBars" option has not been ticked. When zooming in on a screen in Touch Screen systems, it can be panned to view the parts not showing.



Screens cannot be zoomed when their Fit In Window option has been enabled.

### Advanced: Use Intellisense

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

## Screen Execution Properties

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

### Close Screen Delay

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

### Monitor Number

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

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

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

**MultiMonitorMap** item.

Default value:

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

Value changed for exchanging monitor 1 with monitor 2:

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

Practically, the map defined a series of pairs separated by '|' in which the first element represents the monitor configured as the property in the Movicon.NExT project and the second element represents the actual destination monitor in the system. Great care must be taken when modifying the file of interest and to avoid any mishaps from happening we strongly advise you do a backup copy of the ScreenManager.dll.config file before changing it.

### Keep Always in Memory

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



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

### Advanced: Load Objects Synchronized UI

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

### Advanced: Write Timeout

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

### Advanced: Apply Window Settings On Load

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

### Advanced: Load Symbols Delay

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

### Advanced: Force Writing On Server

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

## User Access

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

### Login Required

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



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

### Users for Visibility and/or Access WebClient Side

In this property it is possible to insert one or more Users (separated by ";", eg. User1; User2) that can view the screen in the TilePage and in the GeoPage from WebClient WPF / HTML5. In this case, the screen will be visible only to the defined Users and not by the remaining Users.

If the project is published on WebHMI, the screen can only be opened by the user(s) indicated in the property. When opening the synoptic, the indicated user must already be logged in, otherwise no popup will be opened for authentication, but an error message will be generated.

If the field is left blank, all users of the project will be able to view the synoptic in the TilePage and GeoPage from WebClient WPF / HTML5 and will be able to open the synoptics from WebClient SVG / HTML5 (WebHMI), subject to restrictions on other properties.

### Roles for Visibility and/or Access WebClient Side

In this property it is possible to insert one or more User Roles (separated by ";", eg Role1; Role2) that can display the screen in the TilePage and in the GeoPage from WebClient WPF / HTML5. In this case, the screen will be visible only to Users included in the associated Roles (Groups) defined and not by users belonging to other Roles (Groups).

If the project is published on WebHMI, the screen can only be opened by the user (s) belonging to the Role (s) indicated in the property. When opening the screen, the user of the indicated Role must already be logged in, otherwise no popup will be opened for authentication but an error message will be generated.

If the field is left blank, all users of the project will be able to view the synoptic in the TilePage and in the GeoPage from WebClient WPF / HTML5 and will be able to open the synoptics from WebClient SVG / HTML5 (WebHMI), subject to restrictions on other properties.

## Connection Settings

The Connection properties allow you to define connection sessions, with the relevant parameters, between Client and Server for Tags used by the Screen resources. These Session parameters will then be used instead of the default Session parameters opened by the Client towards the Server.

### Session Name

This property is used to indicate a private Server-Client connection within which to manage the Tags used by the Screen resources. If a Session is not specified in this field, the Recipe will use the entire project's default Session.

### Refresh Time

This parameter is only managed when the Screen resource's 'Session Name' property has been set. This value, which the Client passes to the Server when the Session is opened, determines the time with which the Server sends Tag update notifications to the Client.

### In Use Tags Update Time

This parameter is only managed when the Screen resource's 'Session Name' property has been set. This value, which the Client passes to the Server when the Session is opened, determines the time with which the Server keeps the Tags put in use by Client updated. For example, this value is then also passed to an eventual Communication Driver and used as the minimum time to execute Tasks.

### Not in Use Tags Update Time

This parameter is only managed when the Screen resource's 'Session Name' property has been set. This value, which the Client passes to the Server when the Session is opened, determines the time with which the Server keeps Tags that are not in use updated but which are still subscribed by the Client. For example, this value is then also passed to eventual Communication Drivers and used as the minimum time to execute Tasks.

### Disable when not in use

This parameter is only managed when the Screen resource's 'Session Name' has been set. When this property is enabled, the 'Inactive Item' status is set when the Tag is not use in the Client.

### Number of Items to be Removed

This parameter is only managed when the Screen resource's 'Session Name' property has been set. This parameter is used to set the number of inactive OPC Items to be removed at each time interval defined in the 'Item Removal Delay' parameter.

### Item Removal Delay

This parameter is only managed when the Screen resource's 'Session Name' property has been set. This parameter is used to define the delay time that when expires all the inactive subscribed OPC UA Items will be removed. The number of subscribed items that will be removed when delay time expires is indicated in the 'Number of Items to be Removed' parameter. The delay time is expressed in seconds.

### Always Use Safe Connections

This parameter is only managed when the Screen resource's 'Session Name' property has been set. When this option is enabled, the Client will try to connect to the Server connections declared safe by the relevant certificates for both local and remote connections. In those cases when the Server does not expose safe connections, the next available connection will be used even though unsafe.

## GeoScada

This section includes various properties specific to geo-localization:

### Map Zoom level Visibility

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

### Screen Latitude Coordinate

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

### Latitude Tag

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

### Screen Longitude Coordinate

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

### Longitude Tag

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

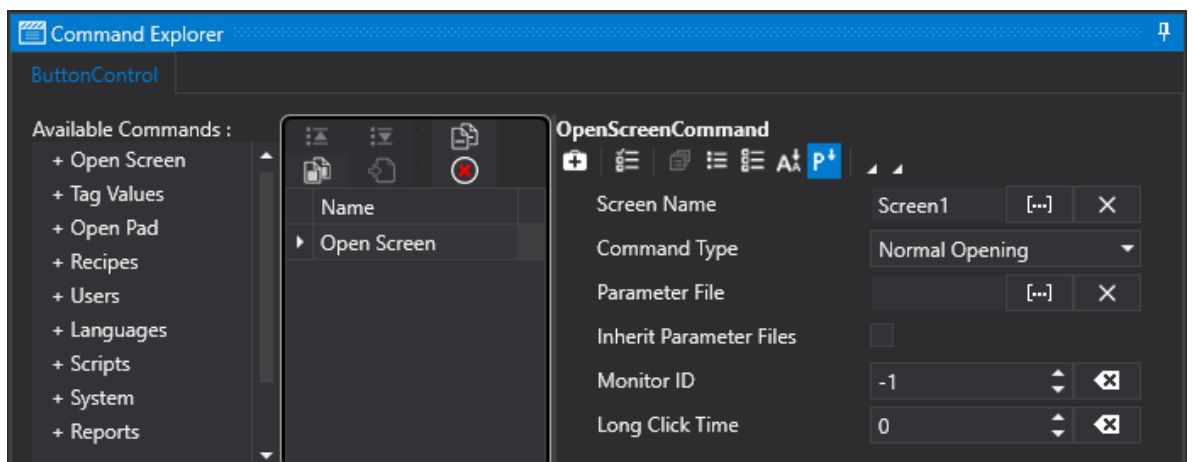
## 6.1.4. Screen Navigation

Navigating screen inserted in the project can be done by using command objects or system command without needing to set any programming command.

## Using change page commands

Usually, change page commands allow you to freely set the navigation between on screen and another. The objects that are normally used for this purpose are Buttons from the Toolbox with the "Open Screen" command. This command executes the closure of the active screen and the opening of the desired screen. Naturally, other command objects and events can be used in the same way (menus, keyboard commands, logic, etc.).

In order to get further information on using command objects, please consult the chapter on "Using Command Objects".



## Using the Project Toolbar and Tile Page

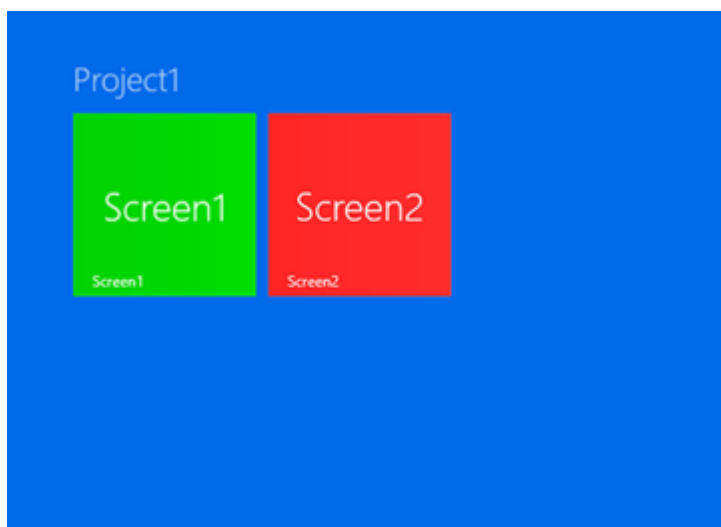
A Movicon.NExT project is predisposed to execute page navigation without obligating the design engineer to insert navigation buttons because the following features have already been configured:

- the use of "Tile Page" at startup
- the use of the project toolbar

### Using the "Tile Page" in startup mode

The "Tile Page" is set for default as startup page.' Nelle proprietà del progetto, è impostato di default il tipo di avvio "Tile Page".

In this way the project will be run displaying a system page representing screens in tile format as the home page.



### Using the Project Toolbar

The "Show Project Toolbar" property is enabled for default in each project screen and permits the toolbar to be shown at Runtime as follows:



This toolbar can be called by placing mouse on the top screen border or by touching the top screen border in touchscreens. The system commands from the Toolbar at Runtime listed below starting from the left of the toolbar:

- Return to Startup Page
- Return to previous page
- Maximise page
- Zoom In - Zoom Out
- Reset Object Manipulation
- Edit parameters
- User Login
- User Logout
- Print

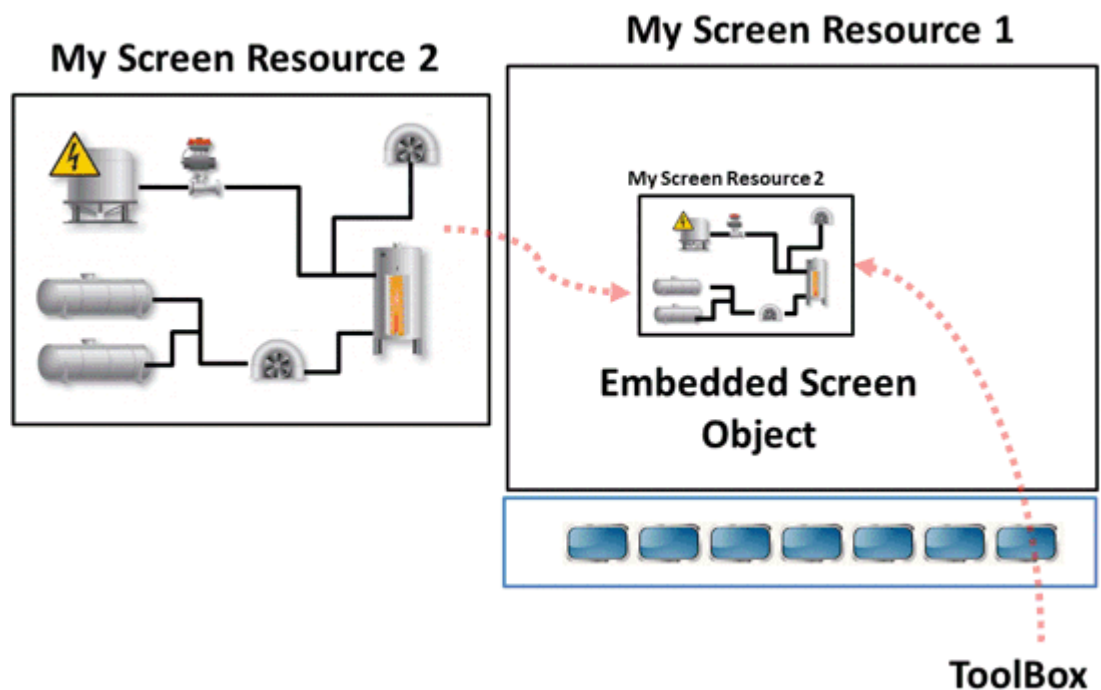
## Using gestures on Touchscreens

Sequential page changes can be done in touchscreen by using the swipe gesture. This type of command has been designed to change those screen pages loaded in memory. Screen pages can be loaded in memory using the "Autoload Screen List" property in Project's properties or they can be kept in memory by setting the "Keep in Memory" property.

### 6.1.5. Embedded Screens

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

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



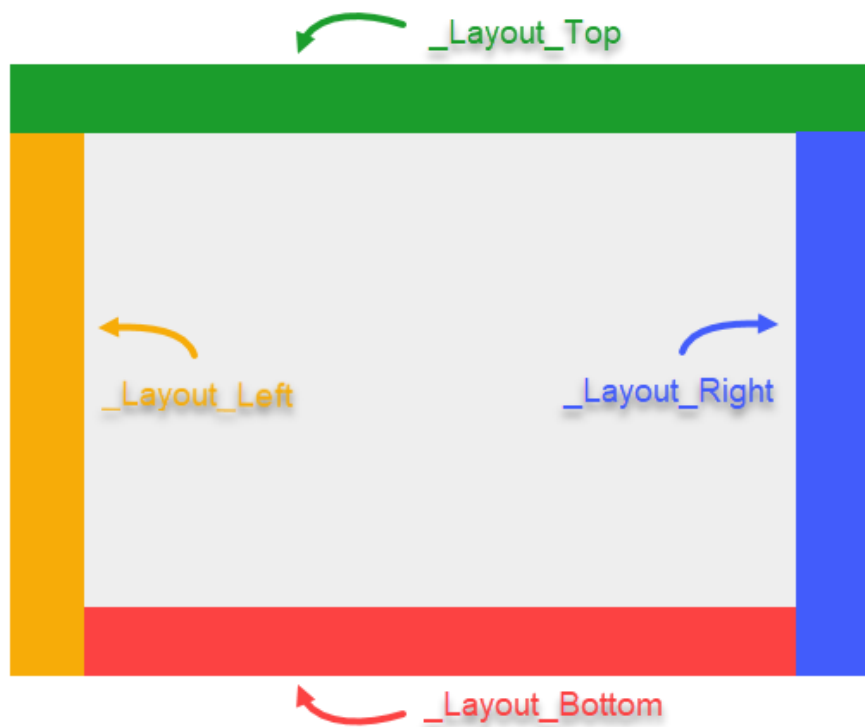
Using Embedded Screens as objects will enable you to have objects that point to screen resources to allow their functions to be called from various parts of the project's user interface, by centralizing the container screen as a "resource".

### 6.1.6. Screen Layouts

User interface Screen Layouts can be created as fixed screen portions that are permanently displayed throughout in the project.

This will allow you to create headings and footers, leaving the remaining space to be used by the screen that you wish to display and which graphic content relating to the context desired.

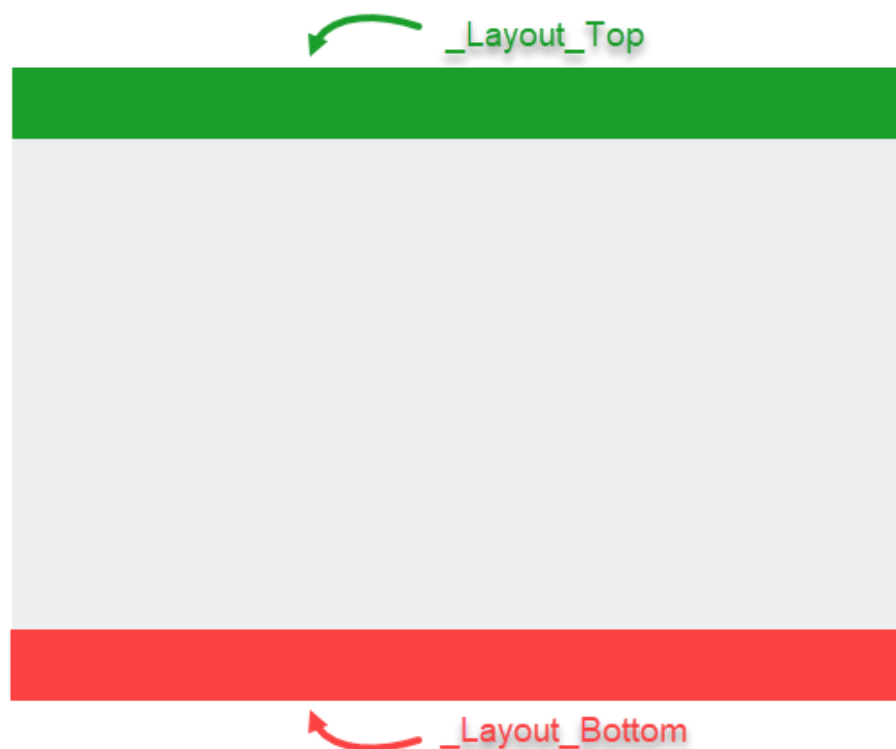
To create screen Layouts, simply assign a pre-established name to the screens as described below.



#### Screen as general Header: `_Layout_Top`

Assigning the special name "**`_Layout_Top`**", Movicon will display the screen with this system name positioned **horizontally** starting from the video screen page's **top left** at runtime.

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





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

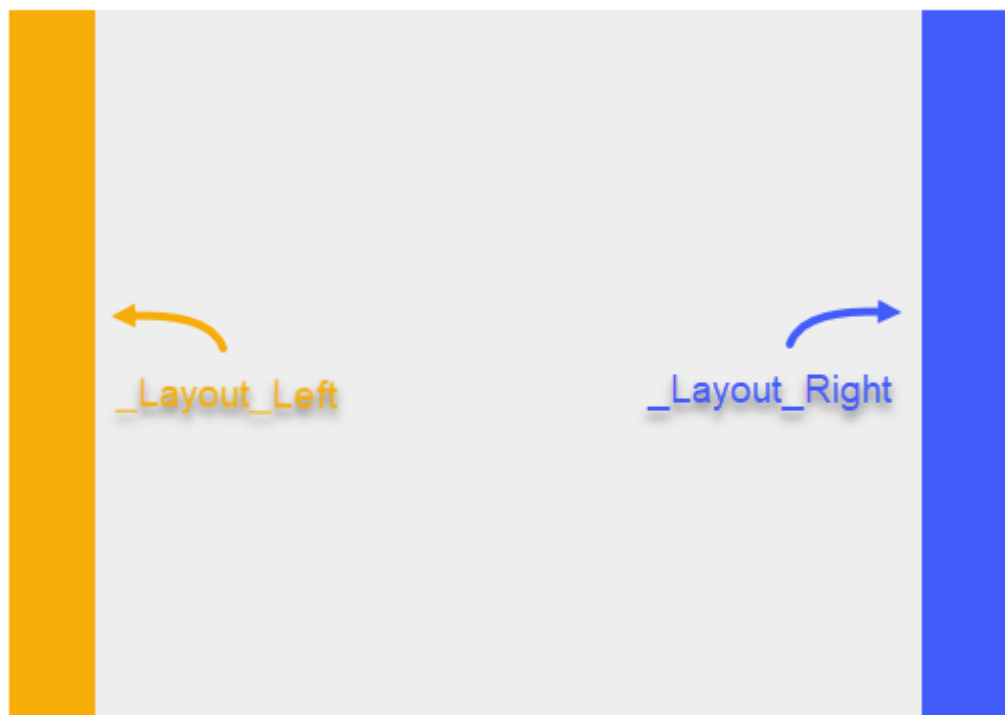
Assigning the special name "\_Layout\_Bottom",In runtime Movicon displays the screen with this system name positioned **horizontally** starting from the video screen page's **bottom left**.

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

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

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

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



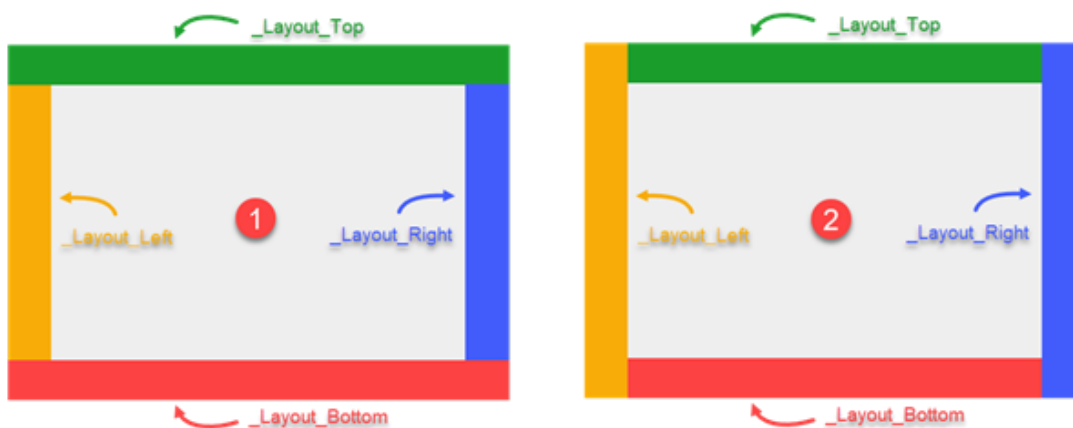
### Screen as general right border: \_Layout\_Right

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

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

### "Vertical Layout Screen Priority" Project property

Enabling this project property will give priority to the "Vertical Layout Screens ("\_Layout\_Left" and "\_Layout\_Right") over the horizontal ones ("\_Layout\_Top" and "\_Layout\_Bottom") as shown below:



- 1) This will be the result when setting this property to False
- 2) This will be the result when setting this property to True

### "Hide Layout Screens" Screen property **Proprietà di sinottico "Nascondi sinottici di Layout"**

Questa proprietà del sinottico, permette di nascondere i Sinottici di Layout quando il Sinottico è aperto. In questo modo si potrà configurare ogni singolo sinottico in modo da mostrare o no i Sinottici di Layout

## Swipe Screens

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

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

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

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

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

### Show Screen to swipe from Left: **\_AppBar\_Left**

By assigning the screen with the special "\_AppBar\_Left" name, Movicon.NExT will show the screen with this name **on the left hand side** of the page at runtime with the auto-show on left to swipe to right of page function ("swipe" touch command). Screen will auto-hide when tapping anywhere on the screen . To call hidden screens without the possibility to use the touchscreen feature, simply click on the right button on the page.

### Show Screen to swipe from right: **\_AppBar\_Right**

By assigning the screen with the special "\_AppBar\_Right" name. Movicon.NExT will show the screen with this name **on the right hand side** of the page at runtime with the auto-show on right to swipe to left

function ("swipe" touch command). Screen will auto-hide when tapping anywhere on the screen. To call hidden screens without the possibility to use the touchscreen feature, simply click on the right button on the page.

### 6.1.7. Pad Screens

#### Pad screen customization

Movicon screens can be used to create customized pads to open in at runtime instead of the standard ones. The following standard windows can be replaced:

- User Login
- Change User Password
- Numeric Pad
- Alphanumeric Pad

These windows can be replaced by those defined with different styles and customized number buttons to be used by the operator.

#### Custom Pad Functions

In order to use Custom Pads you should assign screens that have been disposed with pre-established names. You should also use Local Client variables that also have pre-established names. Both the screens and the variables will have to be created in the project.

The names of screens that can be used as Customized Pads are as follows:

- **\_Login\_User** (User authentication request)
- **\_Verify\_Password** (Change Password request)
- **\_Numeric\_Pad** (Numeric Pad opens showing values clearly)
- **\_Alphanumeric\_Pad** (Alpha Numeric Pad opens showing values clearly)
- **\_Numeric\_PasswordPad** (Numeric Pad opens with password value mode)
- **\_Alphanumeric\_PasswordPad** (Alpha Numeric Pad opens password value mode)

The Local Variable that should be inserted in the project, and which should be defined in the editor's root and not in the folder are:

- **LoginUserVariable** (String variable type in which the User name to be authenticated is inserted)
- **LoginPasswordVariable** (string variable type in which the Password of the User to be authenticated is inserted)
- **LoginVerifyPasswordVariable** (String type variable in which the new User Password is inserted in cases where the old Password must be changed)
- **LoginOldPasswordVariable** (String type variable in which the old User Password is inserted in cases where the old Password must be changed)
- **LoginChangePasswordForce** (Boolean type variable which results set at True when it is mandatory to change User Password. For instance, when Password expires. The variable results False when the operation to change password can be cancelled when, for example, requested by a command from the Command List)
- **LoginRequestUserVariable** (String variable type in which the name of the User to be authenticated is indicated)
- **LoginRequestRoleVariable** (String type variable in which the Role request for authentication is indicated)
- **LoginRequestLevelVariable** (String type variable in which the Level request for authentication is indicated)

- **PadValueVariable** (String type variable in which the numeric or alphanumeric value is inserted)
- **PadMinValueVariable** (Double type variable in which the minimum value or number of characters that can be entered is indicated)
- **PadMaxValueVariable** (Double type variable in which the maximum value or number of characters that can be entered is indicated)
- **PadOK** (Boolean type variable that determines whether the operations must be confirmed (True value) or aborted (False value) when closing Pad)
- **LastError** (String type variable contains eventual error message of invalid value entries)

The strings inserted in cases of error in "LastError" variable can be inserted as String ID in the String table to manage customized and multilanguage texts. The error strings are:

- **AlphaNumericPadMaxLength** (the inserted string exceeds the maximum number of permitted characters)
- **NumericPadNotValid** (the inserted value is not a valid numeric value)
- **NumericPadNotInRange** (the inserted value does not respect the permitted maximum or minimum limits)

When a Pad type screen is created with the above name and used local variable characteristics, this screen will open instead of the one for default when necessary, for example, upon the act of a user Login or upon a Tag change command using the Pad. At this point, if the "PadOK" variable is set to True before closing the Pad screen, the operation will be confirmed, for example the Login of a user or the transfer of the value inserted in the command's target tag. If the "PadOK" variable is set to False before closing the Pad screen, the operation will be aborted.

When using the Pad for User Login in cases where the authentication parameters of a user are incorrect when closing the Pad, an error message will be emitted and the window will automatically re-open to allow a new Login.

When using the Pad for inserting a value, in cases when the value has exceeded the limits upon closing the Pad, an error message will be emitted and the Pad will re-open. Use the script code if you wish the warning message to show and that Pad remains open (an example on how to do this can be found in the predisposed symbols in the Library).

### Commands for managing Customized Pads

The Command List's "Tag Value" group commands can be used in order to use Custom Pads for inserting numeric alphanumeric values. In particular, the following commands are available:

- **Append Value:** appends the "Value to be set" to the actual Tag value by executing a concatenation of strings even when the value is numeric.
- **Remove Value:** removes the last character on the right of the Tag's value even when the value is numeric.
- **Change sign:** this command allows the variable value to change from positive to negative or vice-versa. This command is toggle type, meaning that each time it is executed, it will invert the variable's sign.
- **Append ON-OFF decimal:** this command is used to define whether the "Append Value" or "Remove Value" must be executed on the variable's integral part or decimal part. Naturally, this command has significance only when the variable is floating point type. This command is toggle type, meaning that when it is executed, it will invert the position of entry in respect to the current one.

### Ignore Decimal

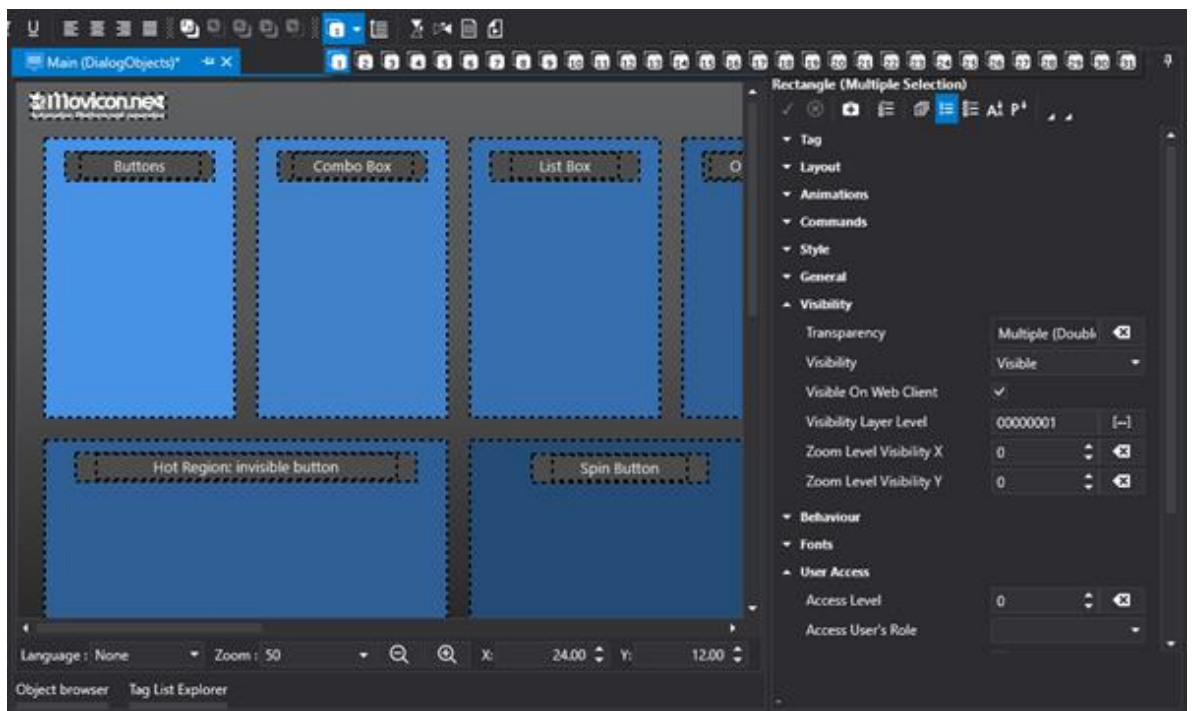
When the "Append Value" or "Remove Value" commands are selected, the following settings will also be made available.

This property is only available for "Append Value" or "Remove Value" command types and when enabled allows the decimal point to be ignored of the value to be edited. In this way, it will be possible to append or remove the "." or "," to the string, by considering them like the other characters. When this property is left disabled, any "." or "," characters (according to the international Windows settings) inserted in the string will be considered as decimal separators.

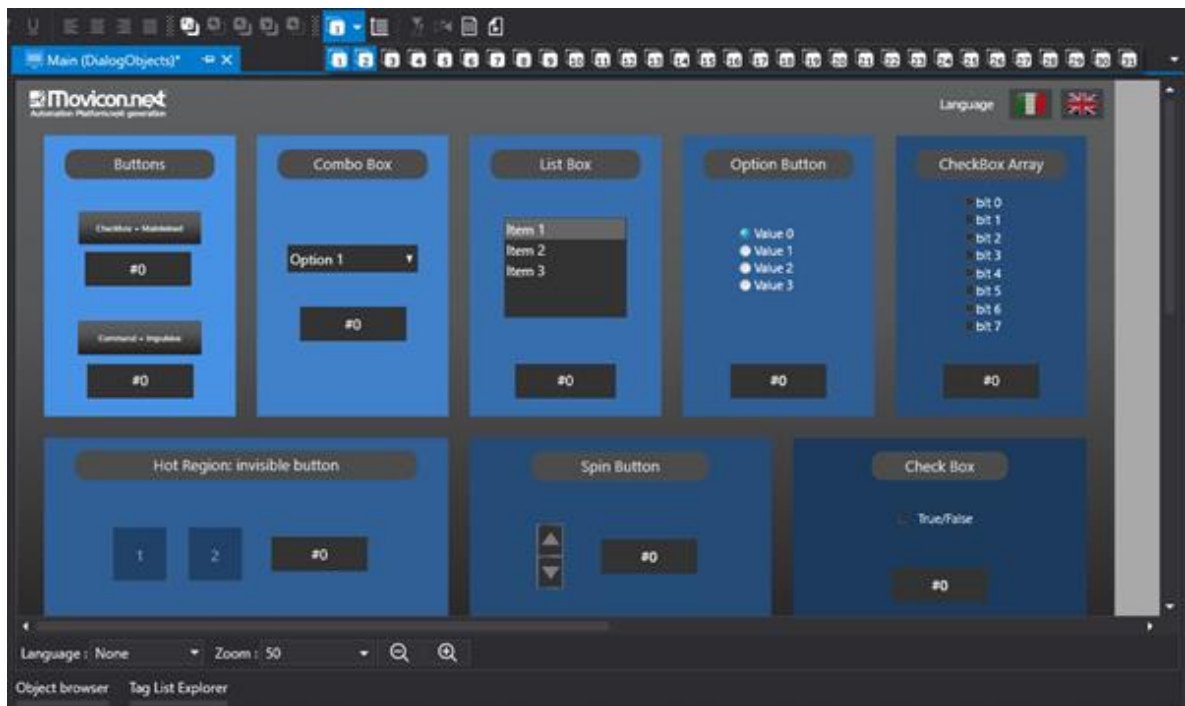
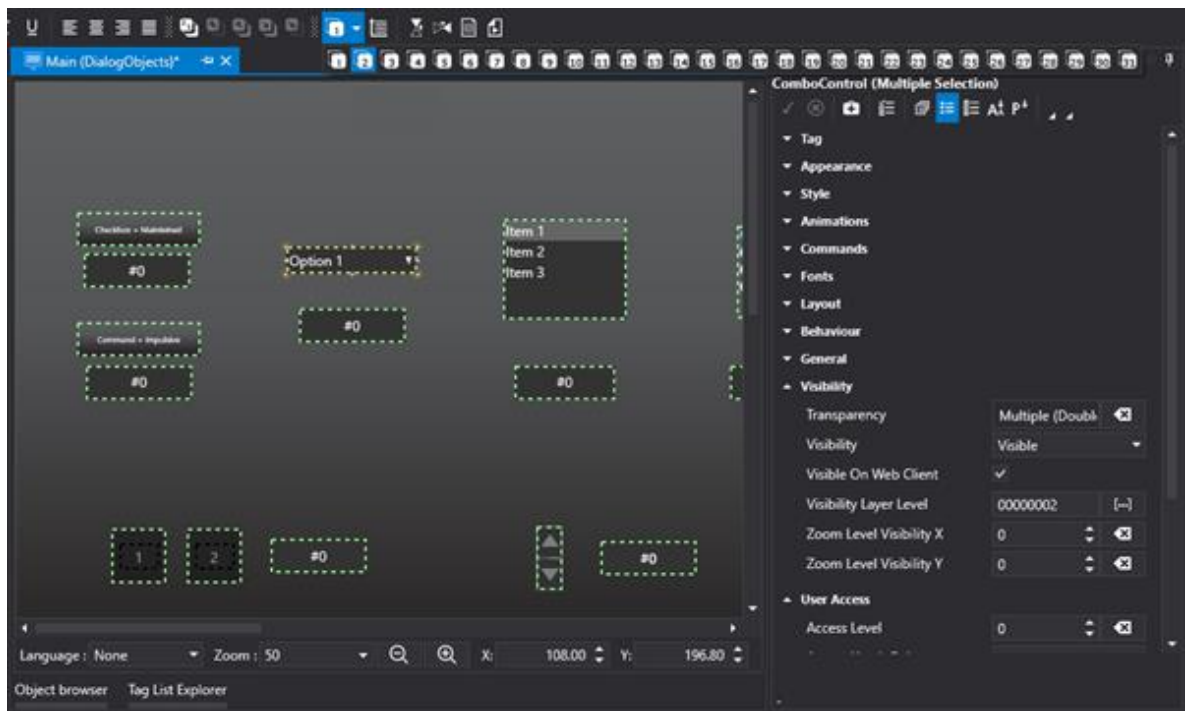
### 6.1.8. Screen Levels

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

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



Once the object level mask has been defined, you can select which level you want to make visible through the "Format> Level> Edit visibility levels" menu or via the shortcut "CTRL + E". window for setting the levels to be displayed. When confirming the settings in this window, only those objects enabled with the same bit mask level as the one defined by this command will remain visible on screen.

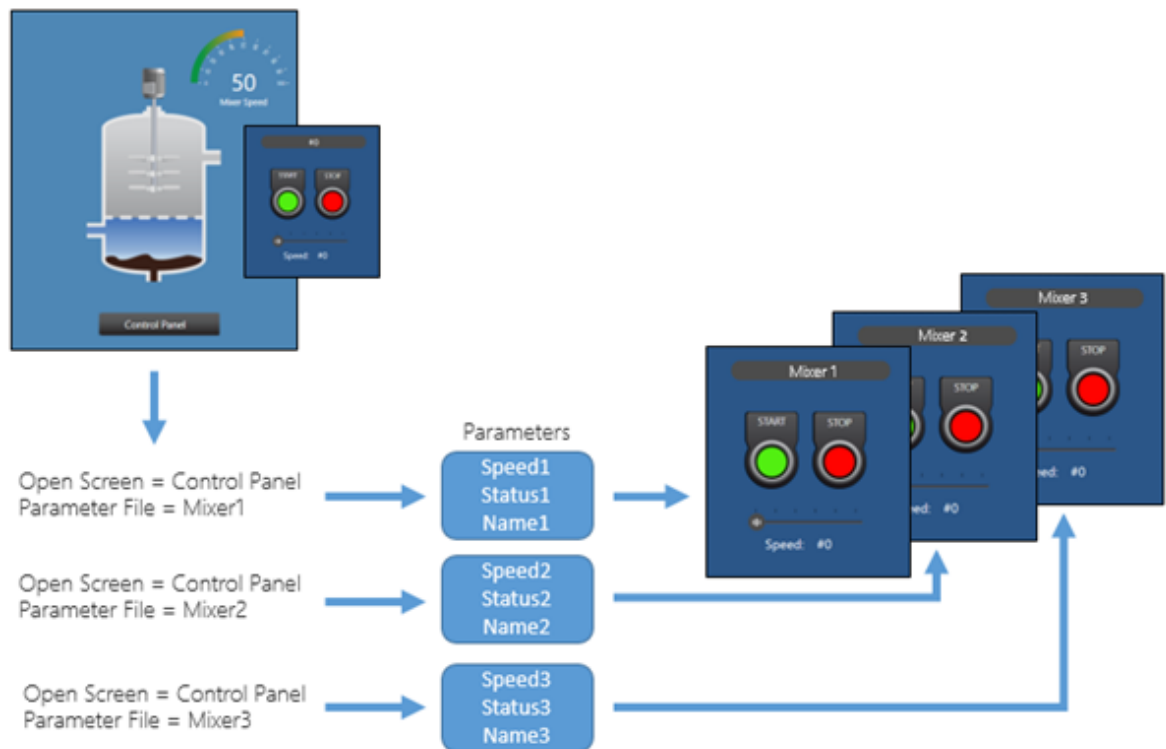


Each screen can be set with a custom visibility level mask so that each screen can display its own levels independently from the other screens. The visibility level mask can be reset when closing the screen so that when the screen is next opened, the mask will be set with all the enabled levels again. The visibility level management only has effect when editing the project but not in Runtime.

### 6.1.9. Screen Parameterization

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

The parameterization techniques applied to screens permit you to create one unique screen that can be called several times with different parameters. In this case, parameters are variables in the Movicon.NEXT Address Space.



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

Movicon.NEXT allows you to address functionalities associated to the display and animation of screen objects and Templates. The programmer will find these functionalities are very handy when creating a project that has a number of screen that are identical graphically but have been associated with different variables.

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

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

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

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



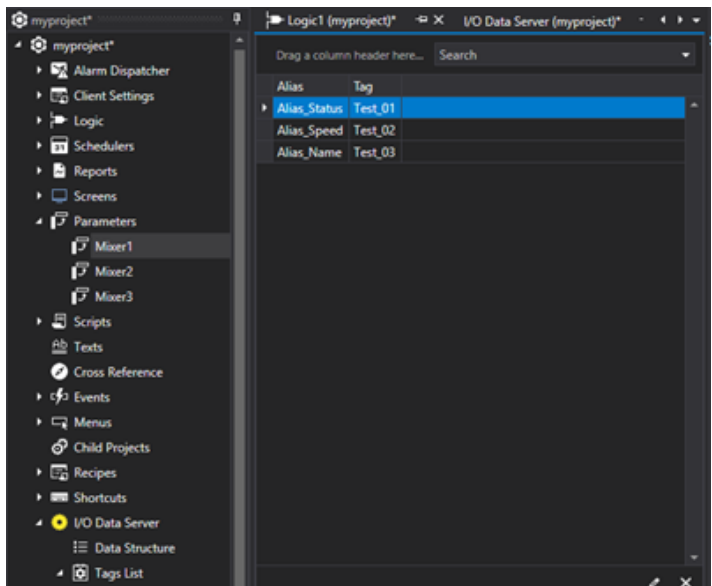
The parameterized screens load the Parameterization File while the screen is being loaded. Therefore, it is crucial that the parameterized screen not already be loaded in Ram. To prevent this from happening you should disable the "Keep Always in Memory" option and set the "Close Screen Delay" property should be set to "zero". Once the



screen has already been loaded, you can however change the associated parameter file by using the "SetParameterFile()" and "ApplyParameterFile()" function scripts.

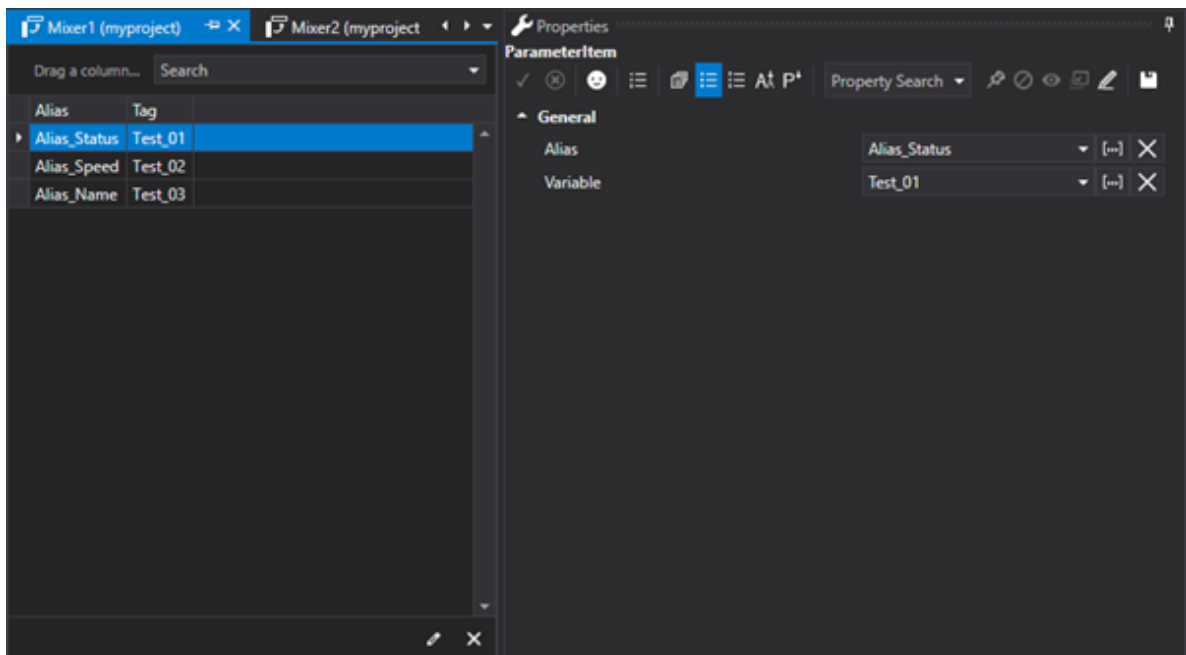
## Parameterization File

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



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

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





where:

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

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



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

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

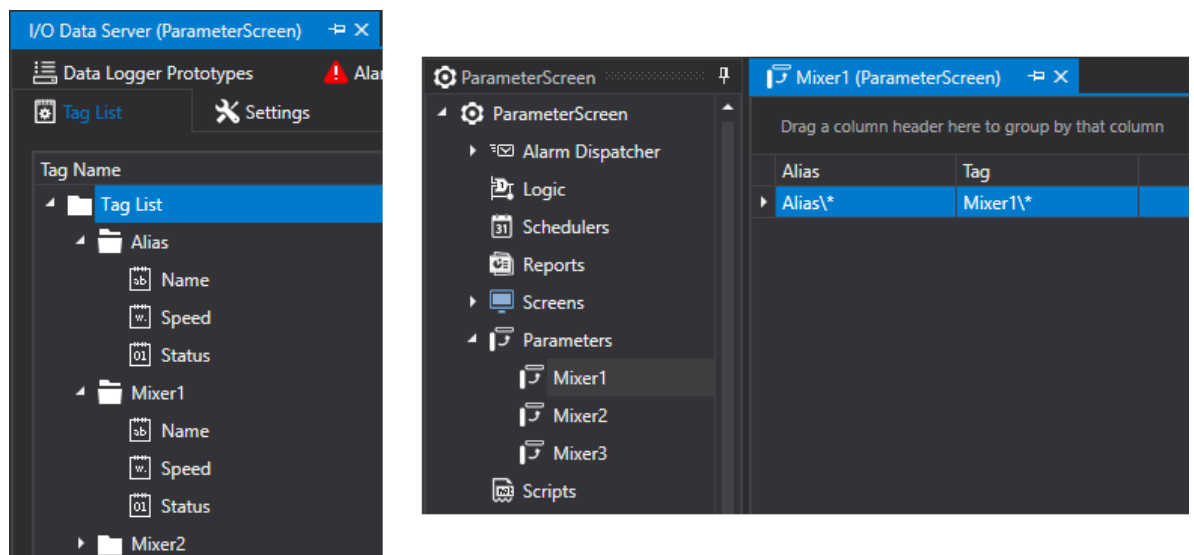


In addition to using the Movicon 'Command List' the parameterized screen can be opened by using the appropriate Basic Script functions as well.

## Using the "\*" wildcard character

It is possible to use the "\*" wildcard character to index an entire folder. In fact, there are structured variables in folders, and each folder contains Tags relating to a specific component. It is possible to significantly reduce the number of parameters to insert in the file. For example, if there are Tags structured as shown in the screenshot below.

It will be possible to insert only one parameter instead of three in the parameterization file like so:



Naturally the structure of the folder that contains Alias Tags should be the same structure of the folder of the Parameter Tags to be substituted with. In this case, any other subfolder of the main folder will be considered for substitution. In addition, this type of indexing functions also for Structure type Tags.

### Parameter Files inheritance

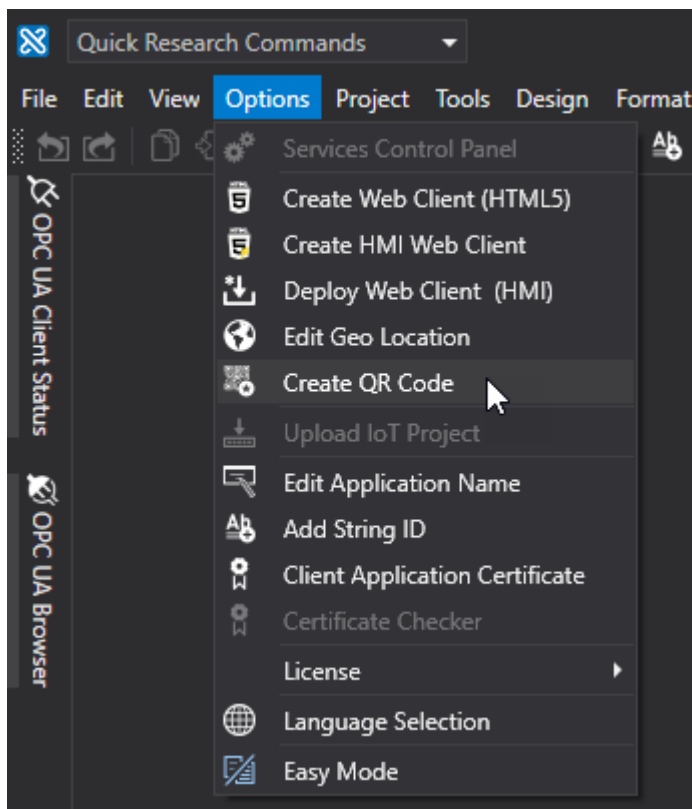
In the opening command of a screen, there is also a "Inherit Parameter File" property that when enabled makes the File inheritable when several screens are open in cascade mode. In practice, when passing a parameter file within one open screen, will open another screen with its "Inherit Parameter File" property enabled and this latter will use the same parameter file passed in the first screen.



The parameter files are automatically inherited by embedded Screens.

### 6.1.10. QR Code

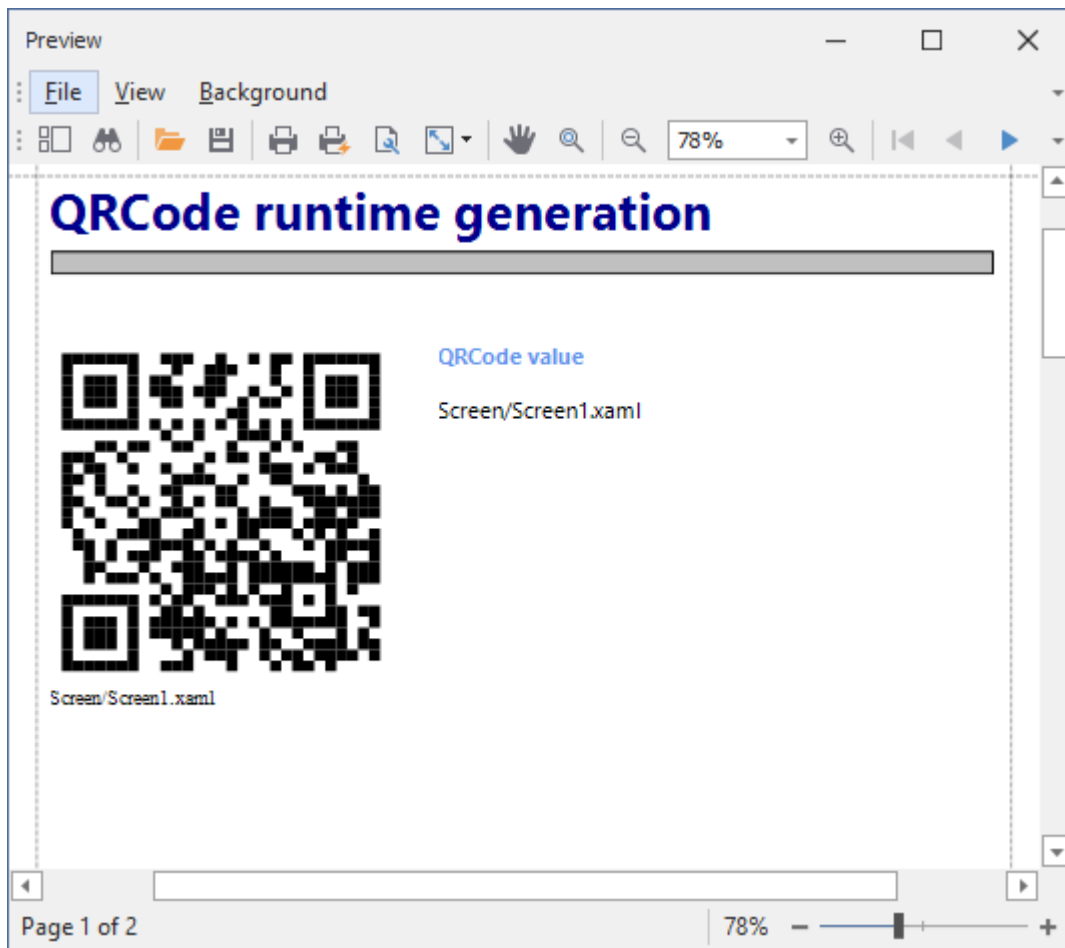
QR Code or Quick Response Code is a 2-dimensional barcode composed of black modules arranged in a square grid. It is used for memorizing information to be read by a smart phone or mobile phone. 7,089 numerical characters or 4,296 alphanumerical characters can be stored in just one cryptogram. The Matrix format is usually 29x29 squares.



After having selected a screen in the Project Explorer Window, it is possible to execute the "Create QR Code" command from the "Options" menu. This will generate a page containing the QR Code relating to the selected screen. The generated page can then be printed or exported in files of various formats.



The QR Code is very handy when used in combination with the Movicon.NExT WebClient APP. As from the 3.4 version, the App can be used to scan QR Code, previously generated by Movicon.NExT and applied to the machine or in certain points of the plant, to open a corresponding screen to save operators time navigating within the project using the app.



### Screen Parametrization with QR Code

The QR Code can also provide access to parameterized screens by adding the parameter file to the name of the screen to be displayed as follows: <ScreenName>,<ParameterFile>.

Any online service, such as [www.qr-code-generator.com](http://www.qr-code-generator.com), can be used to create a QRCode with combined <ScreenName>,<ParameterFile>.

When the screen or parameter file are within a folder, use the "/" char to define the file's path. You do not need to use the file name extensions.

Example 1: MyScreen,MyParameterFile

Example 2: MyScreenFolder/MyScreen,MyParameterFolder/MyParameterFile



The container of the "Screen" project tree must not be included in the screen's name.

### Screen Parameterization with QR Code in AR

To activate this feature in Augmented Reality, you will need to modify the "WebNExTHMISettings" section in the appsettings.json file manually which is archived in the WebHmiServer folder with these options:

- **ARScanQRCode**: The option enables the possibility to recognize a QRCode in combination with the TensorFlow engine.
- **ARScanQRCodeOnly**: This option enables the possibility to recognize a QRcode only and disables the TensorFlow engine.
- **ARScanQRToleranceMS**: This helps the user reduce the flickering between the opening and closing of the popup using the ARScanQRCode option.

## 6.2. Graphic Editing

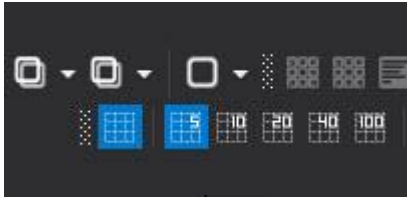
### 6.2.1. Editing Operations

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

#### Using Grids

Editing graphics can be made easier by using grids which are managed by means of using the "Grid" buttons on the screen's layout toolbar or by means of using the "**Design**" menu.

The main **Grid** options are:



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

#### Stack Order

The objects are stacked by overlapping each other in Tab order. The stack order, and therefore the priority in which symbols are displayed, can be changed whenever needed by using the buttons on the screens' toolbar, or by using the "**Visibility Level Editor**" command from the "**Layout**" menu, or ultimately using the 'Z layer' command from the contextual menu which opens with a right click.



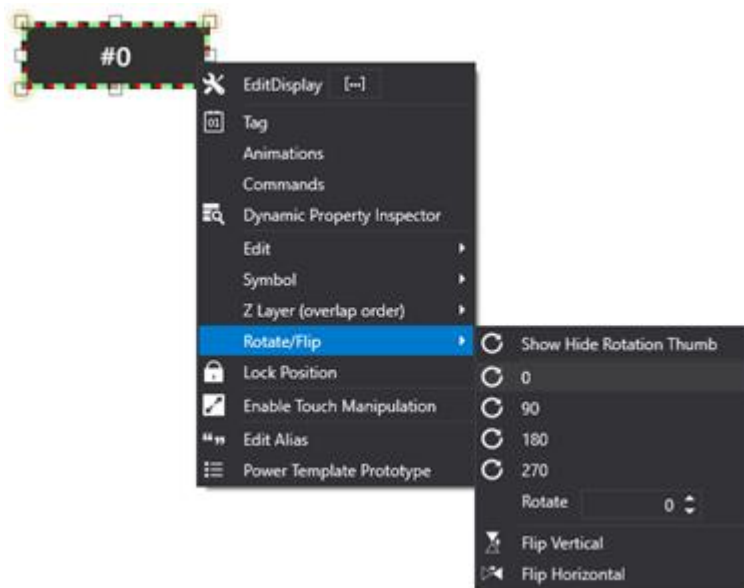
- **Move First:** Sets the stack order of the object selected at first place.
- **Move Last:** Sets the stack order of the object selected at last place.
- **Move Next:** Sets the stack order of the object selected at the next place.
- **Move Prev:** Sets the stack order of the object selected at the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

#### Rotate - Flip

Each object inserted on screen can be freely rotated by clicking on its corners :



or by using the "Rotate" commands from the "Layout" menu or the "Rotate/Flip" command from the contextual menu.



By using the menu you can also:

- Display or hide the rotation center.
- Manage the rotation by degrees.
- Apply a specific rotation angle.
- Flip the object over horizontally or vertically.

## Alignment

Editing groups of object can be made easier by using the Alignment commands or using the toolbar buttons relating to the screen or the '**Align**' command available from the "**Layout**" menu. These command activate after selecting a group of objects and always refer to the **reference object**. The reference object will be the one highlighted in the group of objects. In cases where a multi-selection has been the CTRL+CLICK command, the reference object will be the one in the last group selected and will be highlighted to distinguish it from the rest.



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



The commands below indicated can be sued to align objects in respect to their barycenters:



Note: it is also possible to change the barycenter to rotate the object with: you will need to enable the barycenter point using the "**Show-Hide Rotation Thumb**" command from the menu which shows when right clicking on object. A red spot will show within the object which can then be repositioned as pleased by right clicking on red spot keeping the right key pressed down to move and change the object's rotation point.

## Size

Editing the graphics of a group of objects can also be made easier by using the Size commands which are available by using the screens' relative toolbars or using the "**Size**" command from the "**Layout**" Menu. These command activate after having selected the group of objects and will always refer to the **reference object**. The reference object will be the one highlighted in the group of objects. In cases where a multi-selection has been the CTRL+CLICK command, the reference object will be the one in the last group selected and will be highlighted to distinguish it from the rest.



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

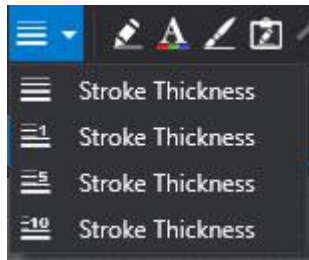
## Spacing out objects

Space Distribution and Alignment of groups of objects can be facilitated by the use of Space Distribution commands, available through the "Placement" command of the Format Menu. This command activates when selecting the group of objects and distributes the selected objects by horizontally and vertically by aligning them evenly into rows and columns using the command's configuration window. Object Alignment is done according to the tab order (overlapping order) in which the objects were selected. Therefore they will be distributed with the first object at the top left, being the object tabbed/overlapped at the bottom, with the last object at the bottom right being the one tabbed at the top.

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

## Managing Borders

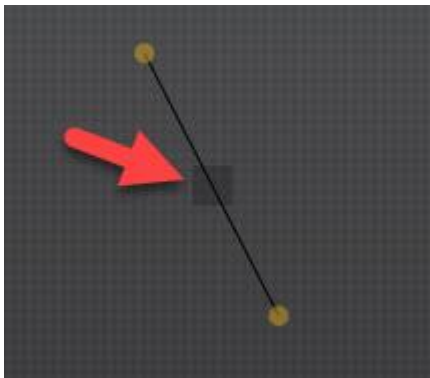
The borders of each object can be sized and set whenever needed using the actual object's properties. They can also be set quicker by using the buttons on the toolbar of the relative screen, or the commands from the **Style** commands from the "**Layout**" Menu.



- **Stroke Thickness** : This is used to set the object's border thickness.
- **Dash Array Stroke** : This is used to set the object's border dash array type.
- **Edit Stroke color** : This is used to set the object's border/back color.

### Positioning

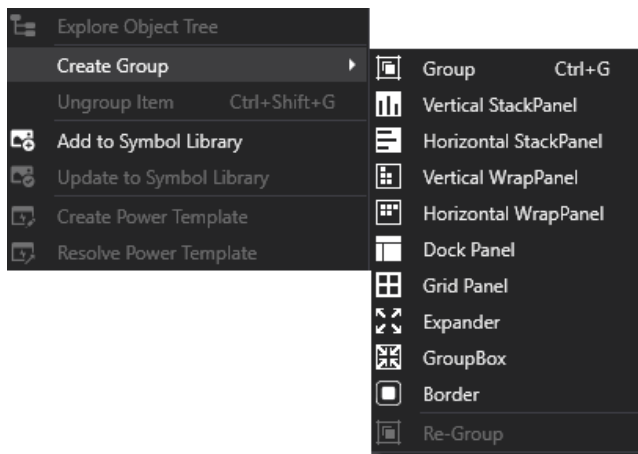
As regards to the positioning objects such as lines or irregular polygons, which have limited moveability given their shape and size, once selecting one of these objects a semi-transparent square will show in the middle. Click and keep pressed the mouse key on the square to drag the object to the position desired.



### Group

When selecting a group of objects, the Group commands can be accessed by using the 'Group' command from the 'Format' Menu accessed from the screen's toolbar, or the contextual menu.

- **Stack Panel**: This is used to horizontally or vertically stack the selected objects one on top of the other, or side by side without touching. This type of grouping is handy when creating list of objects such as button bars.
- **Wrap Panel**: Like the Stack Panel, this is used to stack the selected objects one on top of the other or side by side, but when one object is missing, the other objects will be regrouped to occupy the freed space.



### 6.2.2. Graphics Editor

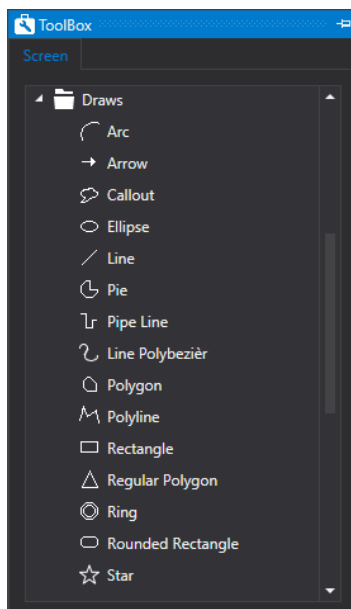
Movicon.NExT has an integrated graphics editor based on the WPF vectorial technology. This editor enables the user to configure screen HMI interfaces by using the primitive draws which are the basic shapes predisposed in the Symbol gallery or the object toolbox.

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

### Graphics editor using basic shapes

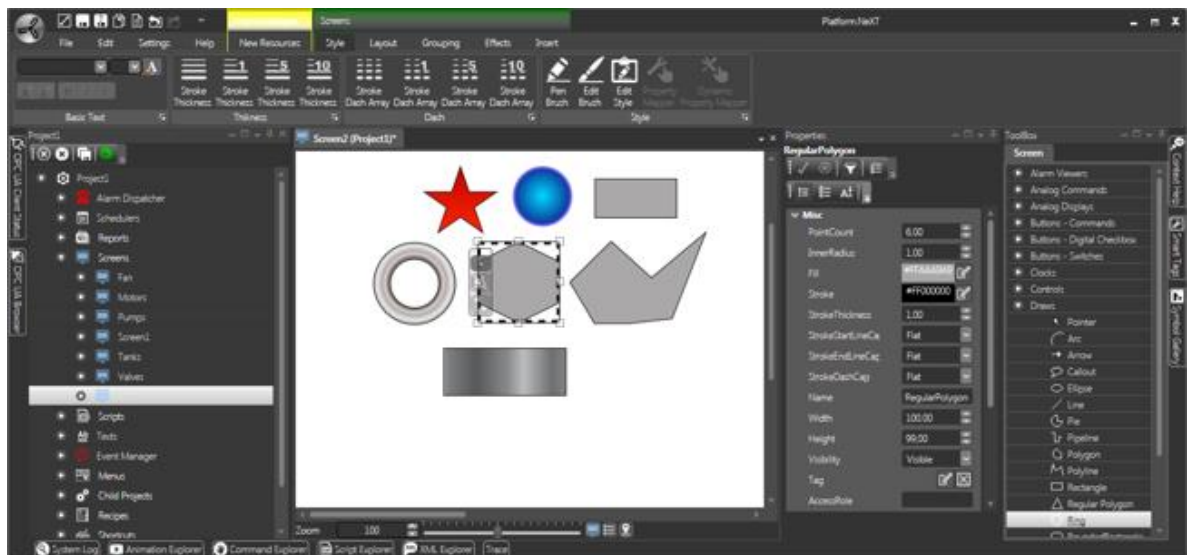
Basic shapes from the ToolBox 'Draws' category can be used to create designs on screen as you please. These shapes can be inserted on screen and configured using the standard techniques common to most vectorial graphics editors.

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



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





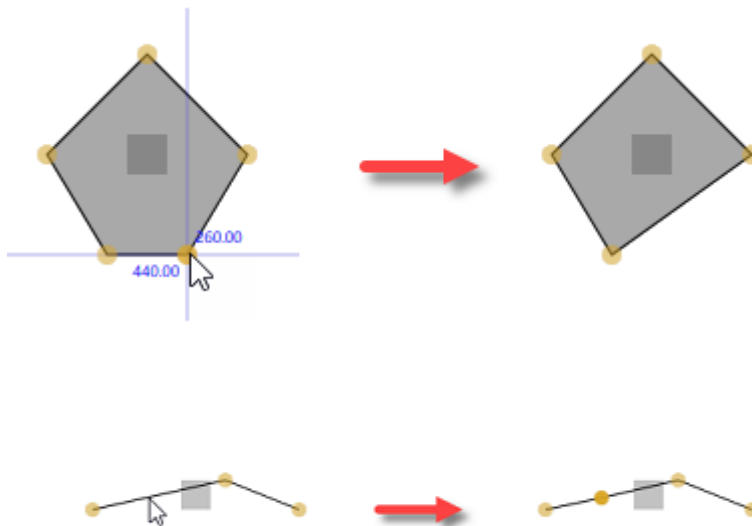
*An example of the Movicon.NExT graphics editor using basic shapes from the ToolBox "Draws" category*




### Editing polygon, poly-line and pipe line points

The element of polygon draws types have intersection points which can be sized and positioned as pleased by clicking on the intersection point wished to be changed and dragging it and dropping in the point desired with the mouse.

To add or remove an intersection point, click and use the SHIFT key at the same time on the interested point as described below:

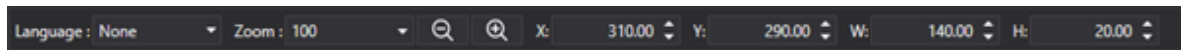
- **SHIFT + CLICK** = Remove/Add an intersection point



-  The basic drawing shapes are also visible in their descriptive XML structure code in the XML Explorer Window. This can be copied and pasted for example in another XAML graphics editor.
-  The basic drawing shapes can be grouped together and saved as Symbols in custom graphics libraries as described below.
-  If the last point of an polyline or pipeline object intersects another object, when you move this object by keeping the CTRL key pressed down, the polyline or pipeline will result anchored to the object and will be moved together with it.

## Zoom

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



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

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

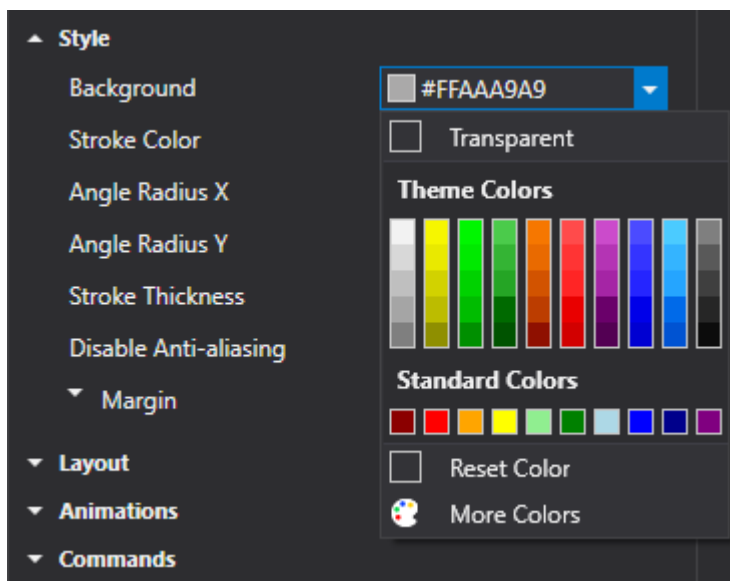
### 6.2.3. Color Selection

Color settings for object borders and backgrounds are done by means of using the Color Selection Window.

There are many different types of color selection settings provided for objects as described below

## Color Selection: Color Palette

Colors can be quickly selected by choosing one of the many colors present on the Color Palette. As an alternative, the Brush Editor can be used as an advanced color selector.



## Custom Color Palettes

The color palette can be customized for the design environment by adding a "ColorPalettes.xml" file name to the Movicon.NExT installation folder ("\_ColorPalettes.xml" file template is available as a template in this folder and can be renamed as "ColorPalettes.xml").

Colors must be defined in the RGB format and this customization will have effect on the Theme Colors and Standard Colors.

The reloading of the Color Palette will be forced when starting up the Movicon.NExT design environment.

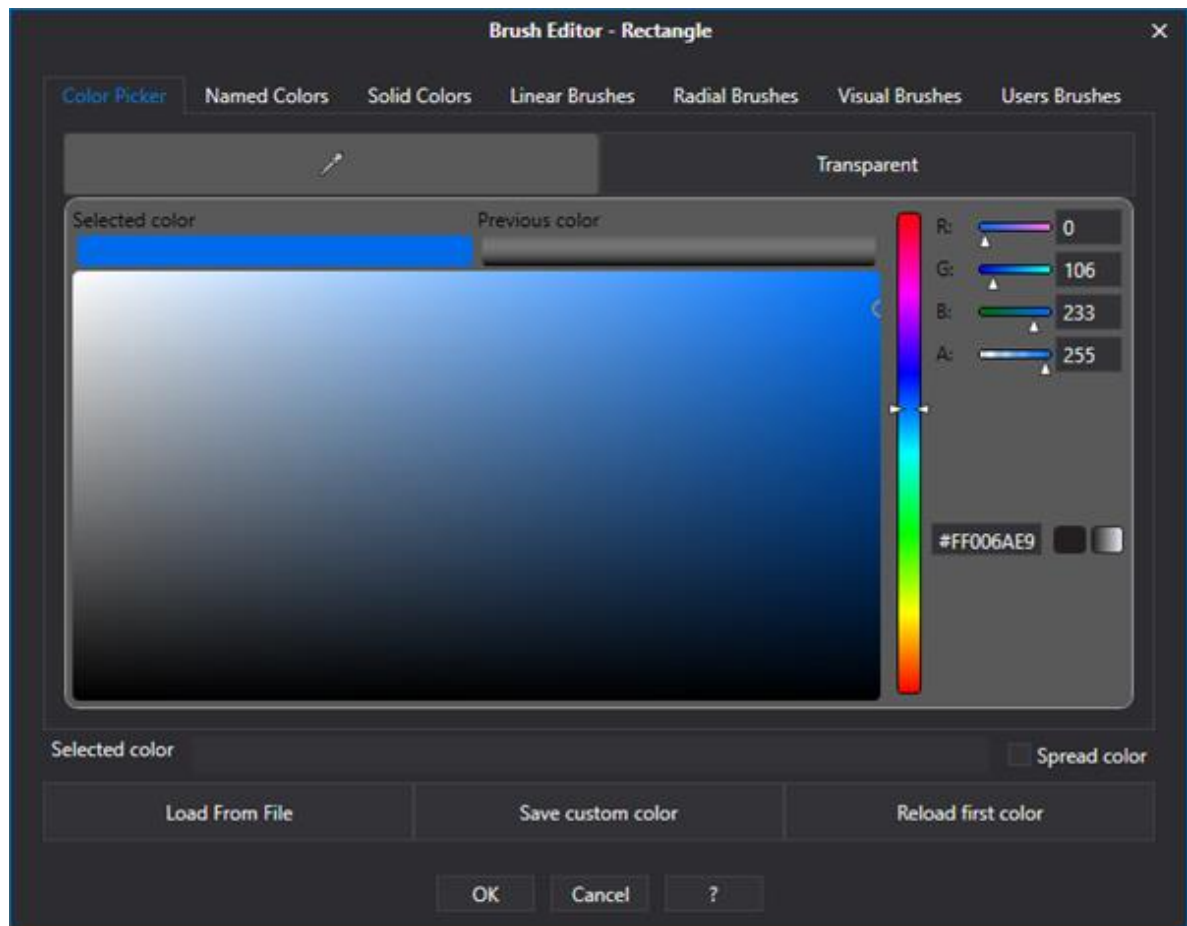
If the file is missing or in the wrong format, the standard palette will be loaded instead.

An example for "ColorPalettes.xml":

```
<?xml version="1.0" encoding="utf-8"?>
<PaletteCollection>
  <Palette Title="First palette" Type="Gradient">
    <Color>255, 255, 255</Color>
    <Color>255, 255, 0</Color>
    <Color>0, 255, 0</Color>
    <Color>0, 128, 0</Color>
    <Color>255, 128, 0</Color>
    <Color>255, 0, 0</Color>
    <Color>128, 0, 128</Color>
    <Color>0, 0, 255</Color>
    <Color>0, 128, 255</Color>
    <Color>0, 0, 0</Color>
  </Palette>
  <Palette Title="Second palette" Type="Standard">
    <Color>255, 255, 255</Color>
    <Color>255, 255, 0</Color>
    <Color>0, 255, 0</Color>
    <Color>0, 128, 0</Color>
    <Color>255, 128, 0</Color>
    <Color>255, 0, 0</Color>
    <Color>128, 0, 128</Color>
    <Color>0, 0, 255</Color>
    <Color>0, 128, 255</Color>
    <Color>0, 0, 0</Color>
  </Palette>
</PaletteCollection>
```

## Advanced Color Selection: Brush Editor

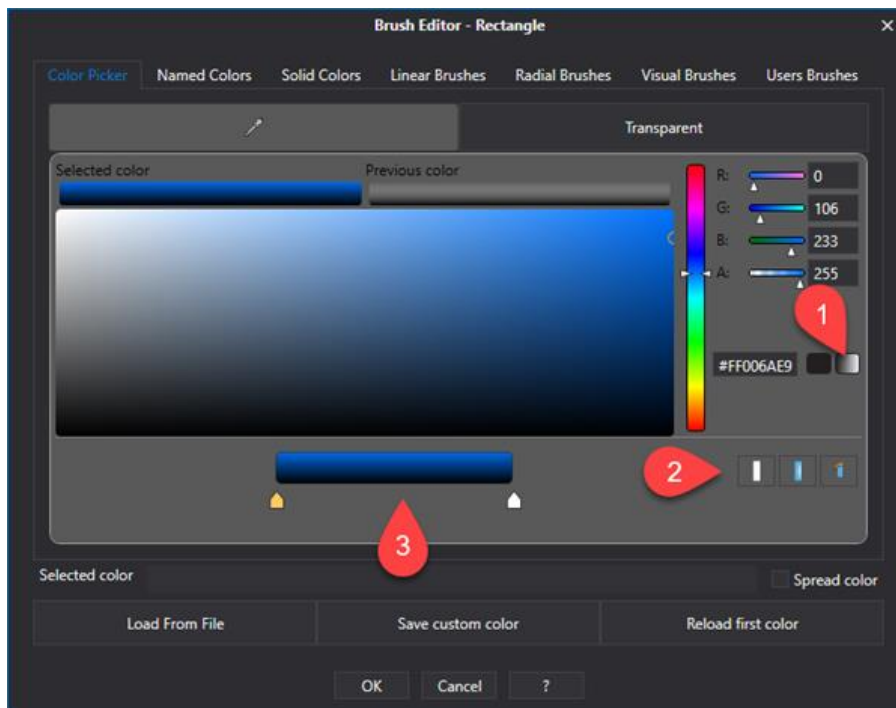
Select "More Colors" from the Color Palette to access the Brush Editor where you can select/define colors using various modes identified by tabs.



- Color Picker
- Named Colors e Solid Colors
- Linear Brushes e Radial Brushes
- Visual Brushes
- User Brushes

#### Tab: Color Picker

When selecting the 'Color Picker' Tab, you will be able to customize a color by using the droplet counter tool (button must be kept pressed and released to pick color) or selecting the color from the Color Table or edit the color code in hexadecimal format #AARRGGBB (Alpha, Red, Green Blue).



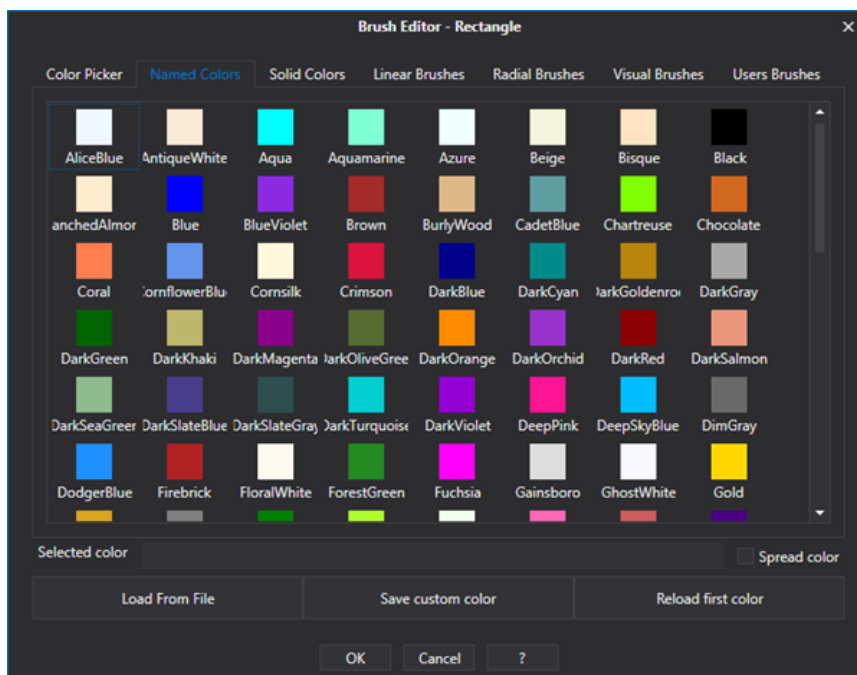
It is also possible to define the color gradients (linear, radial) with two or more colors by:

1. Selecting Gradient mode.
2. Select Gradient type: linear, radial, inverse.
3. Picking each color to use in gradient mode.

The defined color can be save in the "User Brushes" by using the "Save custom color" key.

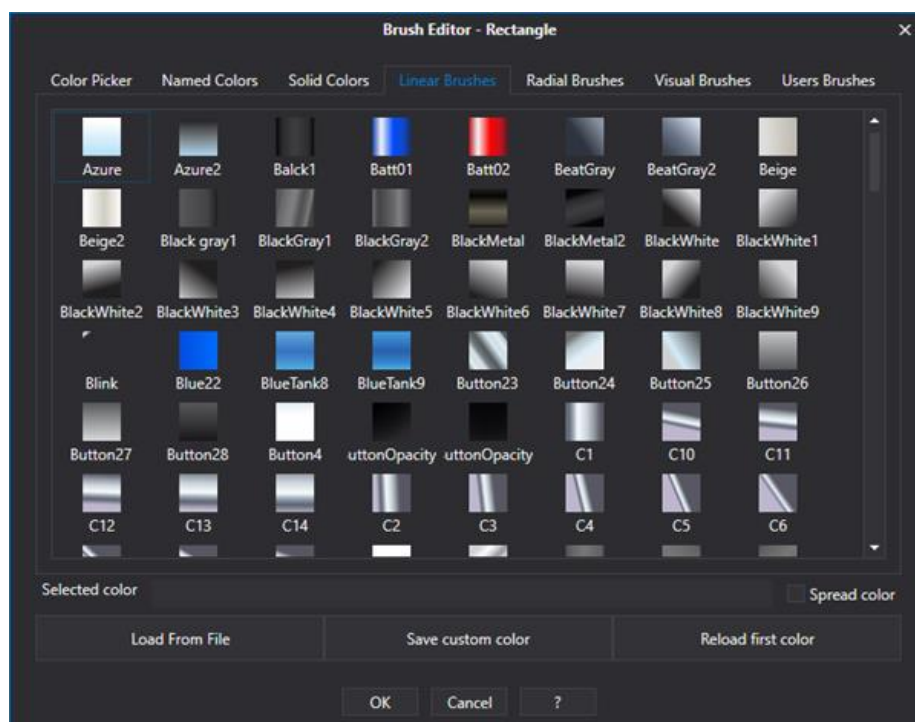
#### Tab: Named Colors e Solid Colors

When selecting the Named Colors or Solid Colors tab, the user will able to choose their favorite color using the solid color palette organized according to two different nomenclature.



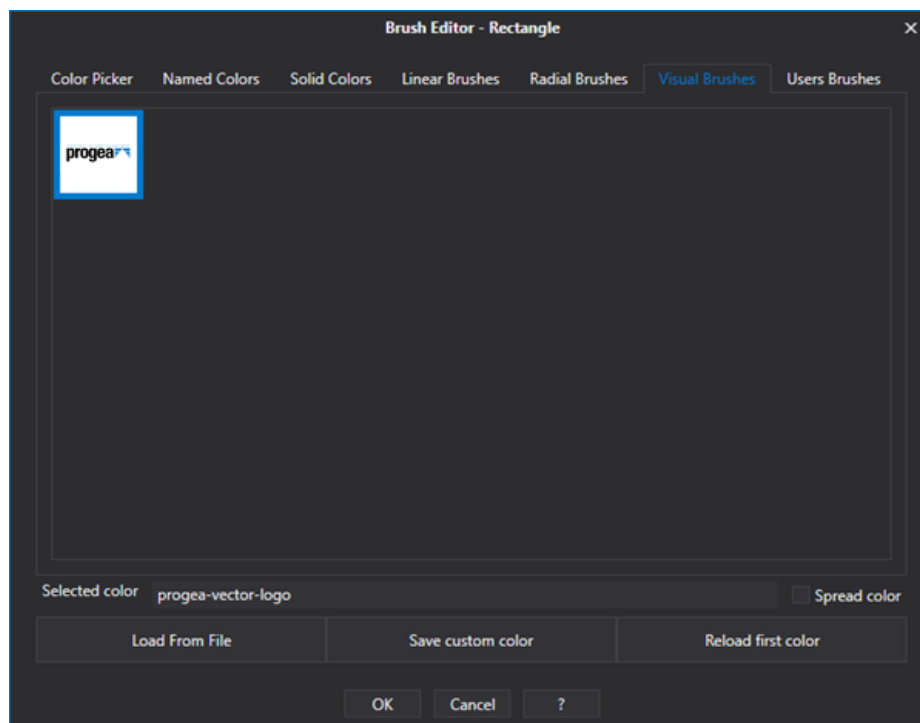
#### Tab: Linear Brushes and Radial Brushes

When selecting the Linear Brushes tab or Radial Brushes tab the user will be able to choose their preferred color using the color palette with linear gradients.



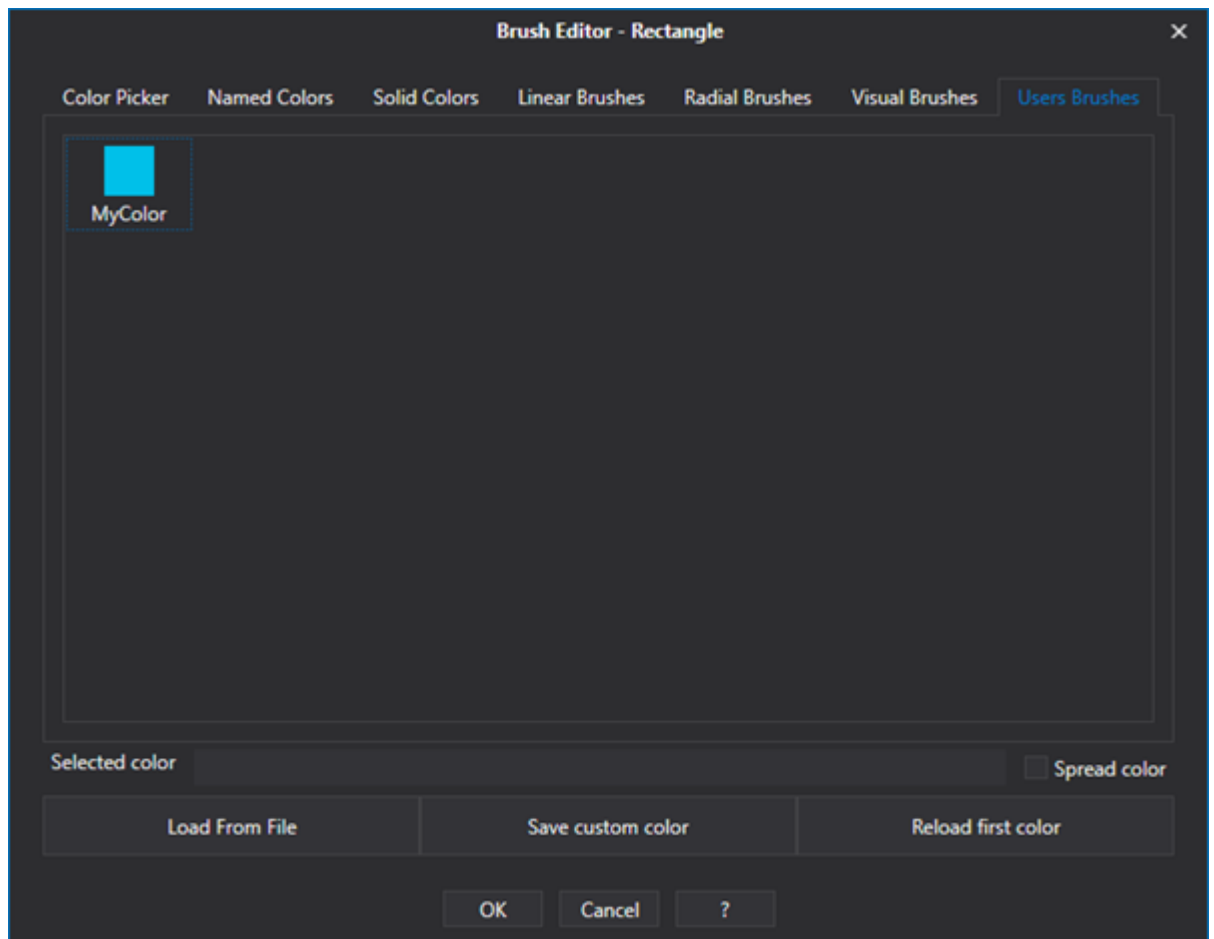
#### Tab: Visual Brushes

When selecting the Visual Brushes tab, the user will be able to choose an image as the background color. The images that are showed in this tab are those in the project's images folder. Other images in .png, .bmp, .jpg, .tif and gif format can also be imported using the "Load From File" button.



#### Tab: User Brushes

When selecting the User Brushes tab, the user will be able to choose one of the custom Brushes that were previously saved using the "Save custom color" button.



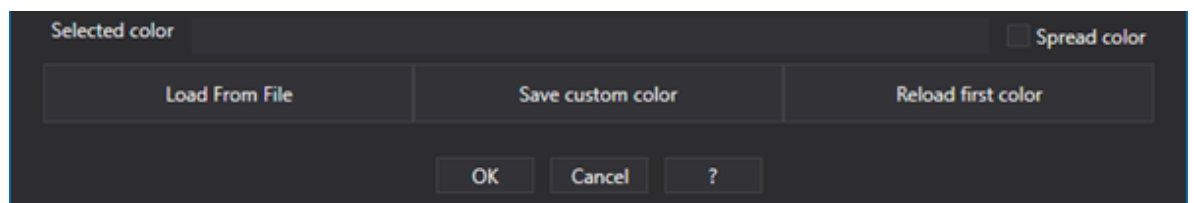
### Other commands: Spread Color, Reload First Color, Reset Color

**Spread Color:** this checkbox allows the selected color to be propagated to all the composed symbol's components.

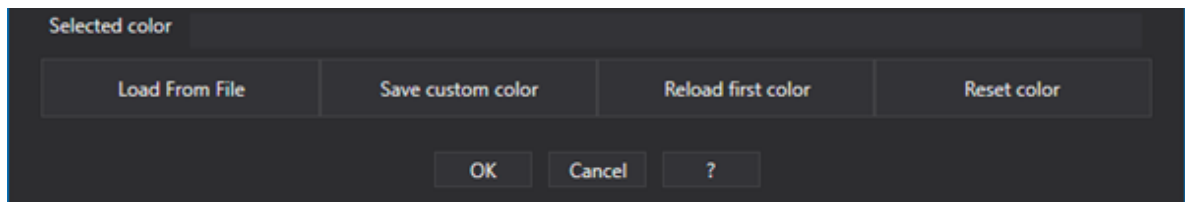


When the background color is propagated to the symbol's components, it will only be applied to elements that have been enabled the possibility to change background color. This command can only be applied to composed symbols.

**Reload First Color:** This command is used to restore the previous color. Basically, this is an "UNDO" of each modification done from the moment the brush editor was opened. This command also has effect on multi-object selections.



**Reset Color:** This command is only available for symbols from the Symbol Library and Buttons with styles from the toolbox and is used to restore the original color. originale.



The three command buttons in this window function as follows:

#### Load From File

This command is used to load an image or video from an external file. Once loaded, the image/video will appear in preview in the "Visuals" Tab and can be selected to associate it to the control. When the image/video is loaded using this command, a copy of the file will be made in the project's "Images" folder.

#### Save Custom Color

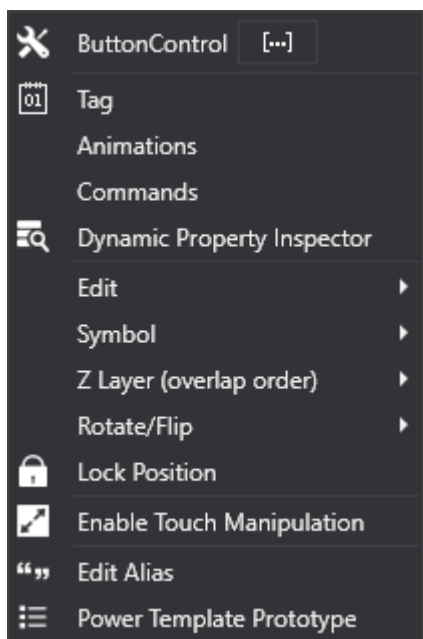
This command saves the custom color that has been created with the "Color Picker" tab for example. The Colors that are saved will then be available in the Color Selection Window's "Custom" tab. These colors are physically saved in the user's IsolatedStorage and therefore will not be made available when moving the project to another machine.

#### Reload first color

This command is used to reload the first color that had control when the Color Selection Window was opened.

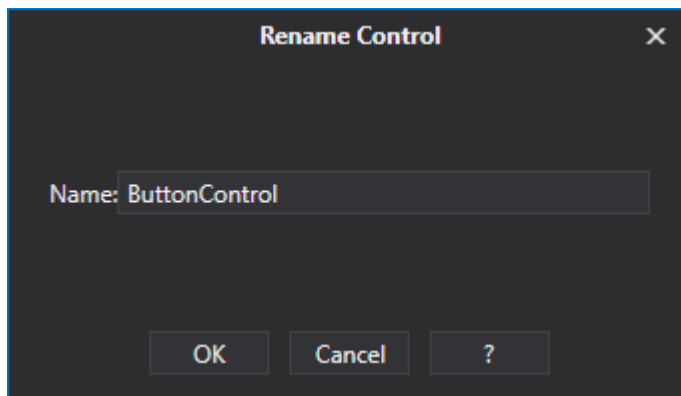
### 6.2.4. Object Contextual Menu

Right mouse clicking on a screen object will open a contextual menu through which it is possible to execute some commands.

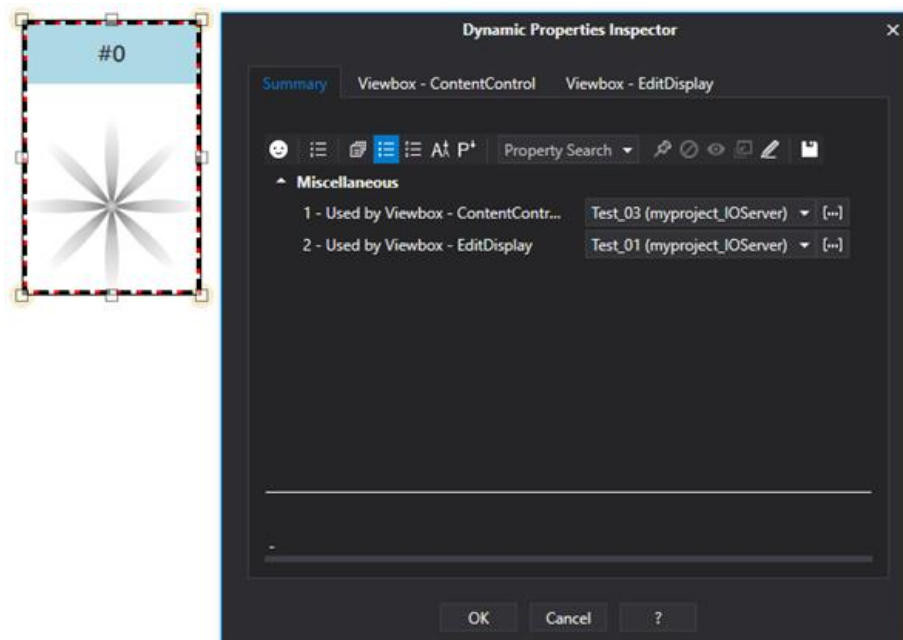


- The first item consent the modification of the object's name that is assigned for default when inserted on screen.





- **Tag:** consents the association of a Tag to the object.
- **Animations:** opens the animation explorer window.
- **Commands:** opens the command explorer window.
- **Dynamic Property Inspector:** opens the window that reports all the tags connected to the object and the objects it is composed of.



- **Edit:** The classical Undo/Redo, Copy/Cut/Paste commands can be used along with some special commands that consent the transfer of the Tag, Animations and Commands.
- **Symbol:** Allows a number of objects to be grouped and customized symbols to be inserted in the project's library.
- **Z Layer:** Permits the overlap order to be changed.
- **Rotate/Flip:** Manages the rotation or Flip over.
- **Lock Position:** Locks the position of the object on screen.
- **Enable Touch Manipulation:** Allows the object to be manipulated at Runtime (for further information please refer to the chapter on <link Multitouch>)
- **Edit Alias:** Edits the object's alias table.
- **Power Template:** displays the Power Template symbol's configuration.

### 6.2.5. Importing bitmap images and videos

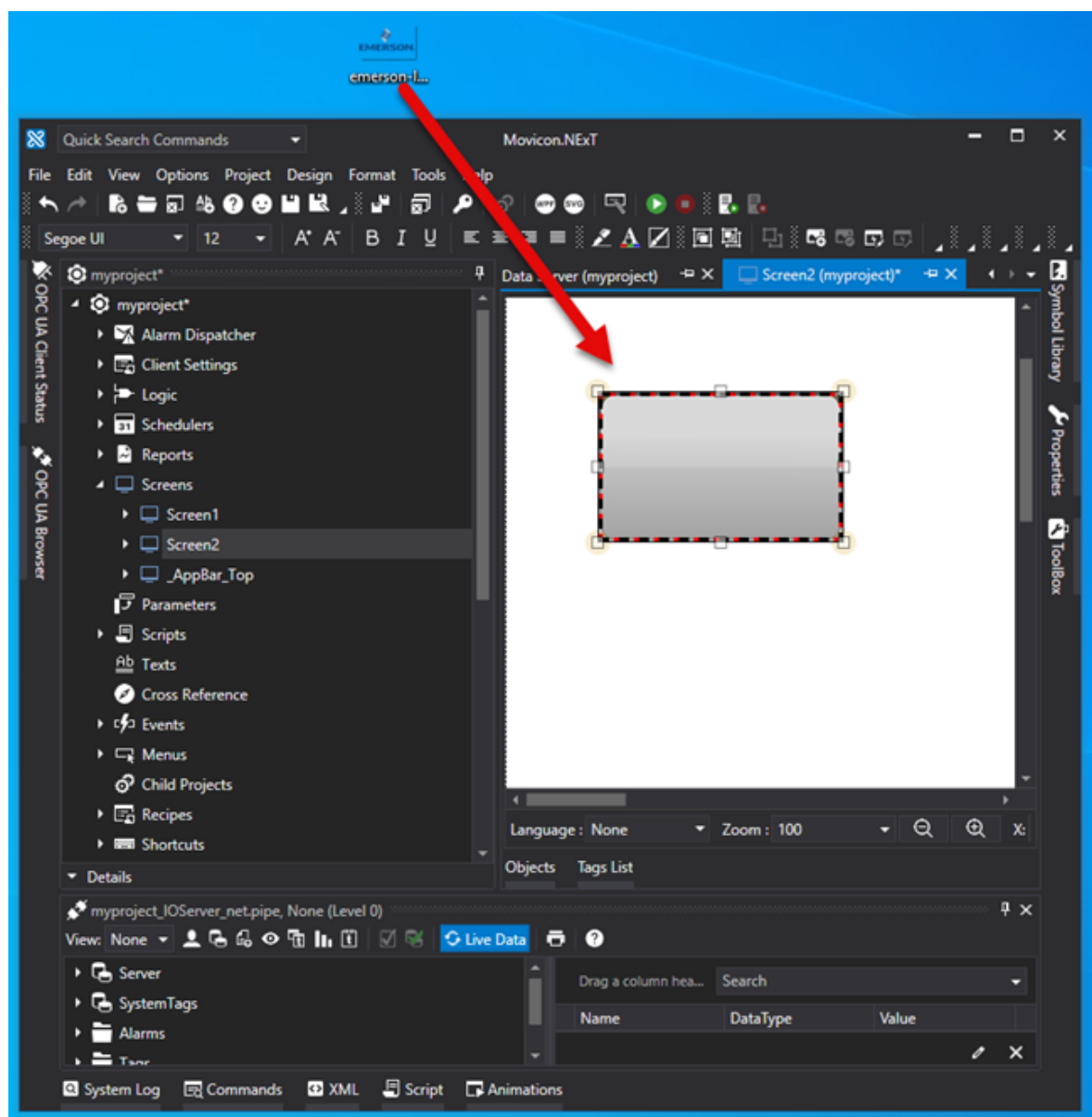
The background of both screens and design graphics can be customized and made multimedia by associating them with external images or video clips.

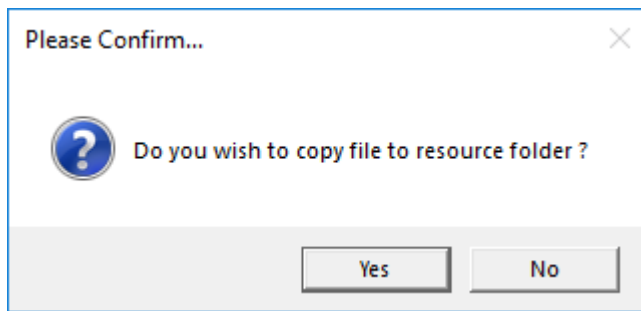
The raster formats supported for the background images are the most commonly used: .BMP, .GIF, .TIF, .JPG, .PNG

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

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

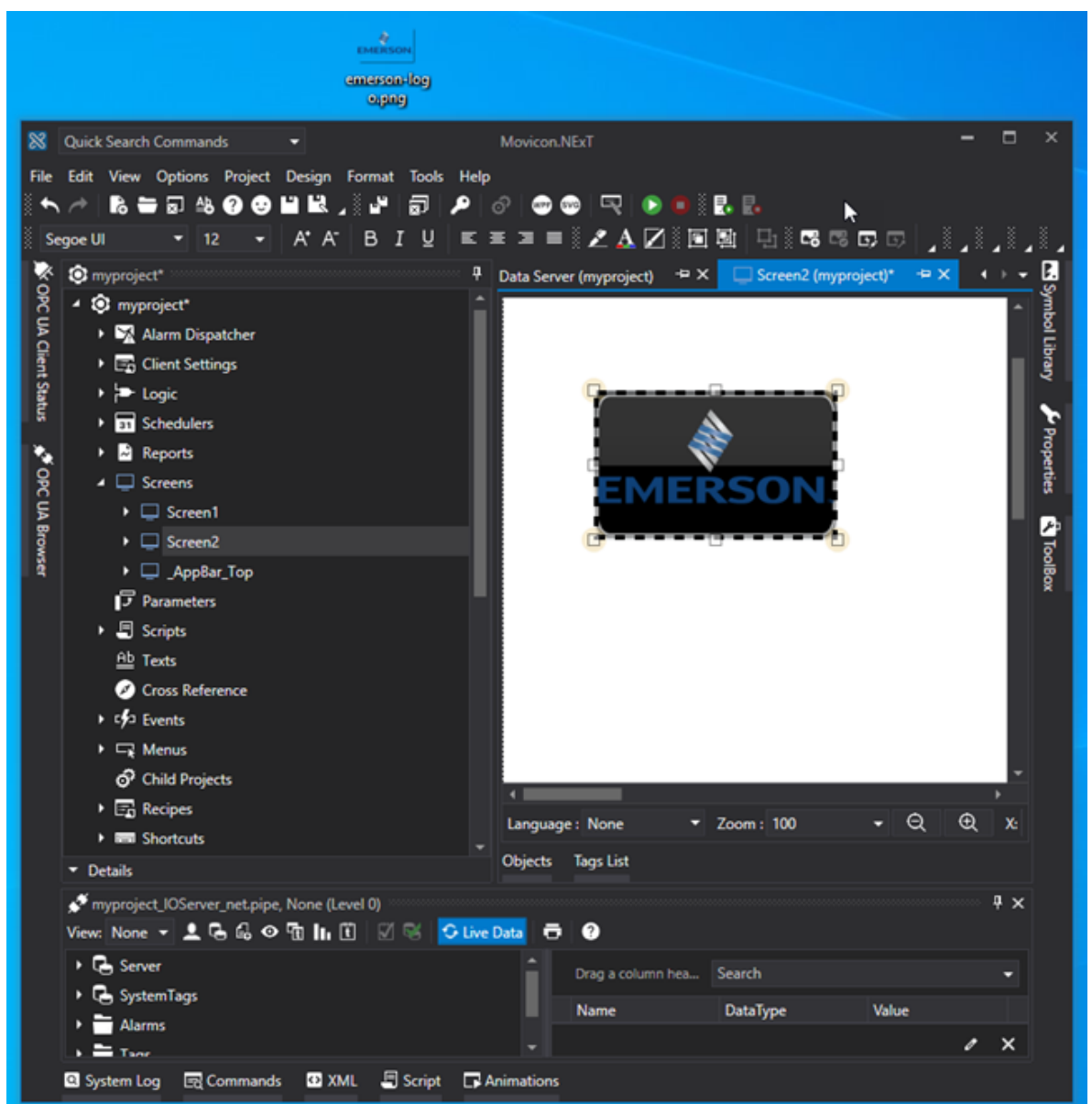
The contents will always be adapted to the size of the object or screen.





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

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



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

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

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

While those supported for movies are:

.WMV, .MPEG, .MP4

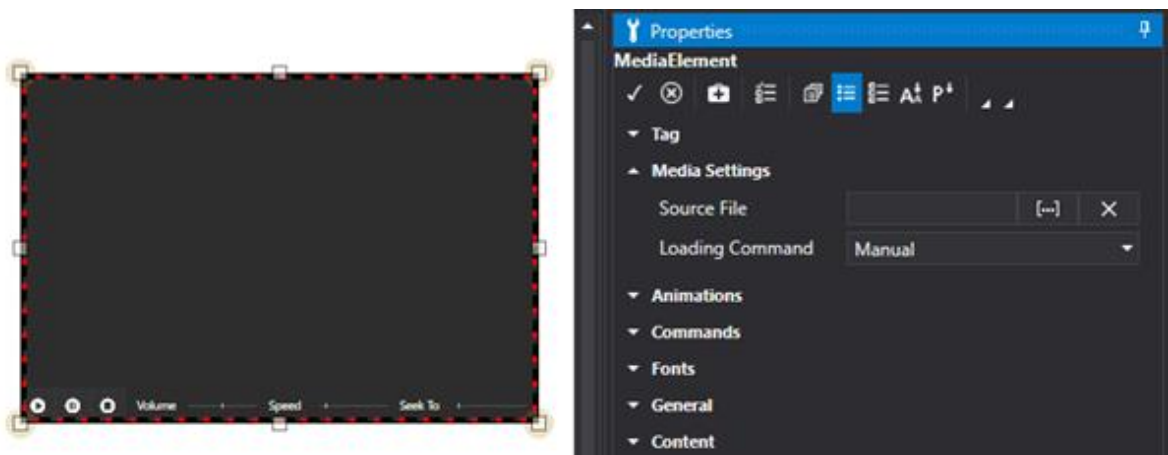


The 'Viewer' objects from the toolbox can be set with a background image or movie by using a property dedicated for this purpose. These objects can be assigned background by using the drag&drop techniques.

The same explanation given for importing bitmap images also goes for videos in formats: .WMV, .MPEG, .MP4



Drag&drop is not supported in the MediaElement or Video Viewer objects from the Toolbox where it is possible to set the video to be reproduced using the Source File property.



### 6.2.6. Importing vector images

The Movicon.NExT graphics engine is based on the modern WPF technology and as a consequence the graphical elements are based on XAML.

While creating applications based on WPF, the eXtensible Application Markup Language (XAML) present in the .NET Framework permits you to describe user interfaces by means of an XML document in schematic mode.

#### Movicon symbol XAML code

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

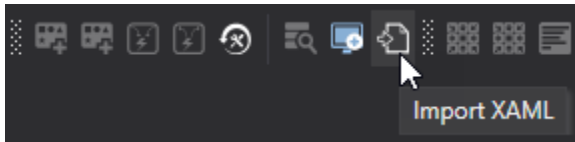


Please contact the Sales Offices or distributors in order to get more information on the symbol library's XAML code protection.

### Importing XAML vector graphics

Movicon.NExT consents to importing vector graphics in XAML format produced by other editors externally. Therefore, third party editors (Microsoft Blend for Visual Studio, Adobe Illustrator, Inkscape) can be used as well as the work of graphic designers to obtain symbols with great graphical impact that can be used seamlessly in Movicon.NExT screens or for creating custom symbol libraries.

By using the "Import XAML" command available in the Toolbar, or the command from the "Tools > Import XAML" menu, it will be possible to select the desired graphics file and import it within a screen.



Once the symbol has been imported on screen, you will need to add it to the project's customized library using the "Add to Symbol Library" command from the object's contextual menu under the Symbols item.

### Converting vector images to XAML

As previously mentioned, in order to import a vector symbol to Movicon.NExT, it must be saved and converted into the XAML format beforehand.

If you are using the Adobe Illustrator, you can install the XAMLEXP for free (<http://www.mikeswanson.com/xamlexport>) which consents you to export the Adobe Illustrator project in XAML format.



If you are using the Inkscape open-source editor (<http://inkscape.org>) instead, simply use the "File > Save As" command and select the "Microsoft XAML (.xaml)" format.



This software can also be used to simply convert a vector symbol from the SVG format to the XAML format!

### Rendering XAML vector symbol dynamic

Movicon.NExT offers the possibility to modify (during Design or Runtime mode) the coloration of imported XAML objects. To create and select the specific parts of an object to be subjected to color variations, you will need to follow some fundamental XAML programming rules. Let's now look at how this is done by using the below image:

# 4

By using an advanced text editor (e.g. Notepad++) or a KAXAML editor (<https://github.com/thinkpixellab/kaxaml>) it is possible to view the source code which will be presented like the one in the image below:

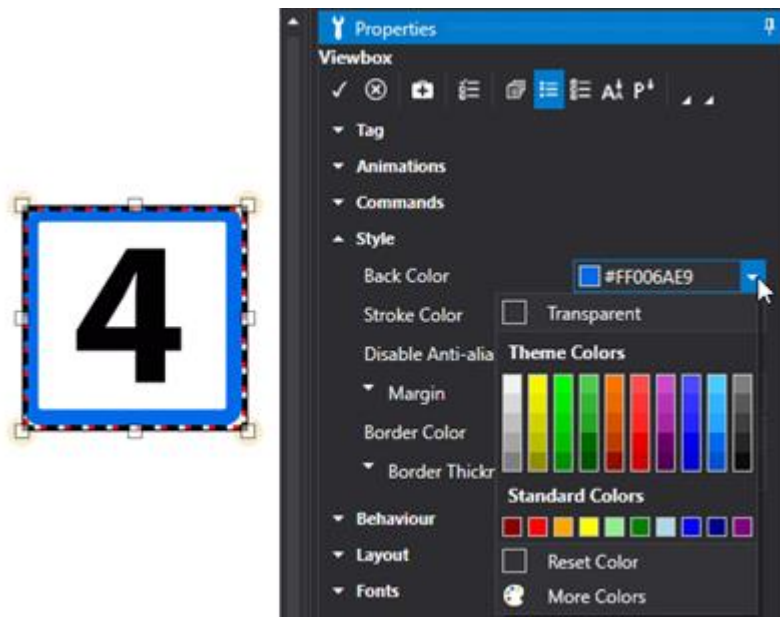
```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <Viewbox xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" Stretch="Uniform">
3  <Canvas Name="icon4" Canvas.Left="0" Canvas.Top="0" Width="15.594" Height="15.594">
4    <Canvas.RenderTransform>
5      <TranslateTransform X="0" Y="0"/>
6    </Canvas.RenderTransform>
7    <Canvas.Resources/>
8      <Path xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Name="border" Fill="#FF030104">
9        <Path.Data>
10         <PathGeometry Figures="M14.252 0H1.341C0.602 0 0 0.602 0 1.341v12.911c0 0.74 0.602 1.342 1
11         </Path.Data>
12       </Path>
13       <Path xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Name="value" Fill="#FF030104">
14         <Path.Data>
15         <PathGeometry Figures="M10.147 3.063h-2.71L3.853 8.832v1.441h4.195v2.258h2.099v-2.258h1.13
16         </Path.Data>
17       </Path>
18     </Canvas>
19   </Viewbox>
20
```

The "paths" are the elements that compose the image. To make the color of an element dynamic, simply insert the Tag="Background" attribute within the chosen path as seen in the below image:

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <Viewbox xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" Stretch="Uniform">
3  <Canvas Name="icon4" Canvas.Left="0" Canvas.Top="0" Width="15.594" Height="15.594">
4    <Canvas.RenderTransform>
5      <TranslateTransform X="0" Y="0"/>
6    </Canvas.RenderTransform>
7    <Canvas.Resources/>
8      <Path Tag="Background" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Name="border" Fi
9        <Path.Data>
10         <PathGeometry Figures="M14.252 0H1.341C0.602 0 0 0.602 0 1.341v12.911c0 0.74 0.602 1.342 1
11         </Path.Data>
12       </Path>
13       <Path xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Name="value" Fill="#FF030104">
14         <Path.Data>
15         <PathGeometry Figures="M10.147 3.063h-2.71L3.853 8.832v1.441h4.195v2.258h2.099v-2.258h1.13
16         </Path.Data>
17       </Path>
18     </Canvas>
19   </Viewbox>
20
```

It is important not to touch the surrounding code, otherwise this will compromise the whole object. Therefore, be careful with the spacing between one character and another. At this point, once the object like the one indicated above has been imported, it will be possible to change the color of the elements to which the attribute has been added by means of using the Back Color property.

The object can receive a back color from the Movicon.NExT editor or can be animated dynamically with the Back Color function.



### 6.2.7. Static Graphics Effects

Each object inserted on screen can be associated with graphical effects that can be selected from the Screen Effect Toolbar or from the Format > Effect menu.



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



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



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



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

#### 6.2.8. Develop and integrate WPF User Controls in Movicon.NExT

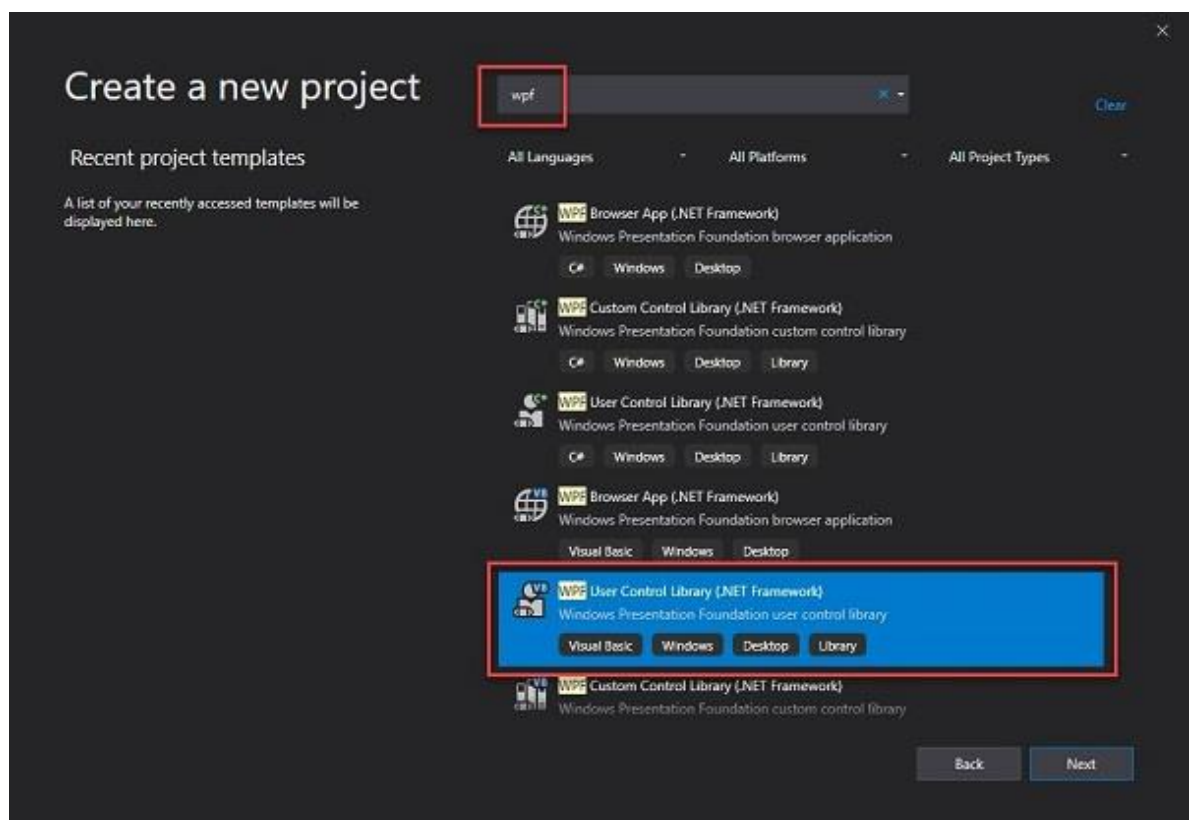
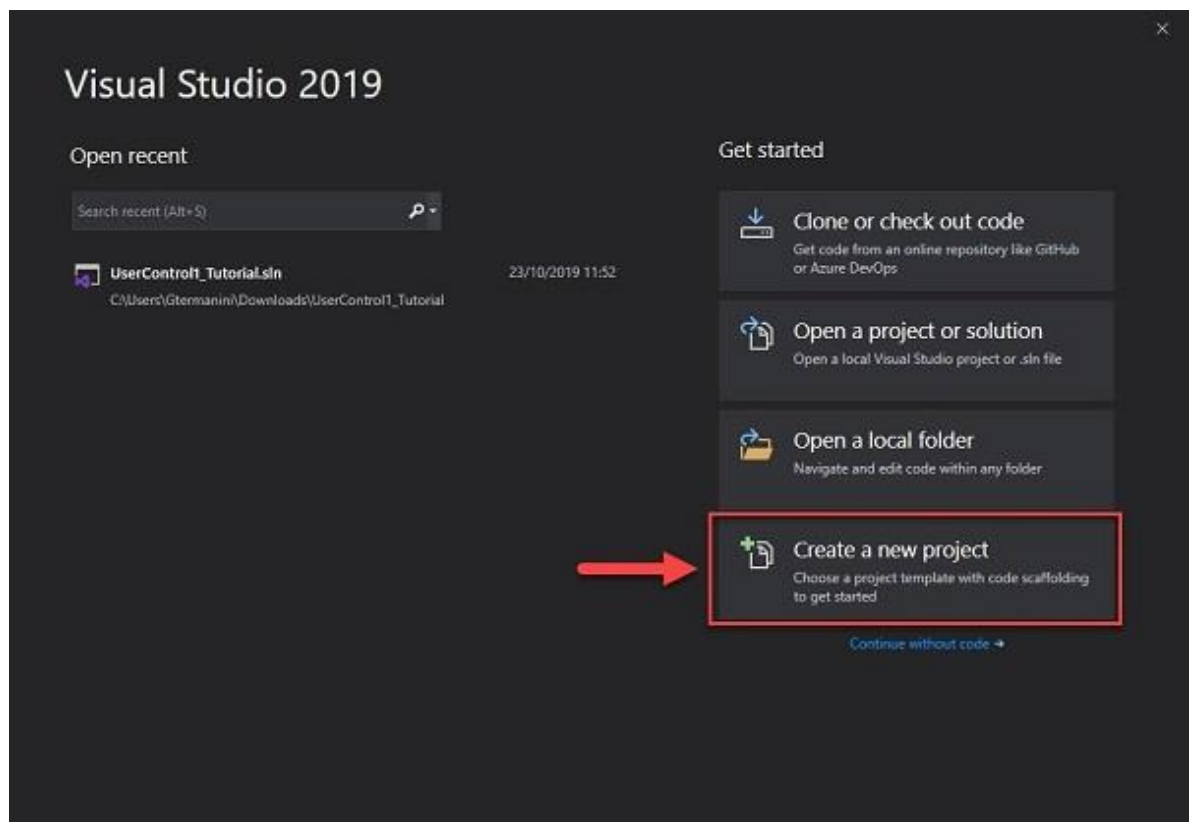
This example shows how to create a WPF UserControl object in the Visual Studio 2019 development environment and how to use it in Movicon.NExT.

### Creating a WPF User Control

By means of using Visual Studio 2019, or its equivalent, you will need to create a new **“WPF User Control Library”** project in which to insert the controls and features desired following these steps below.

Open Visual Studio and create a new project by selecting **“WPF User Control Library (.NET Framework)”**:



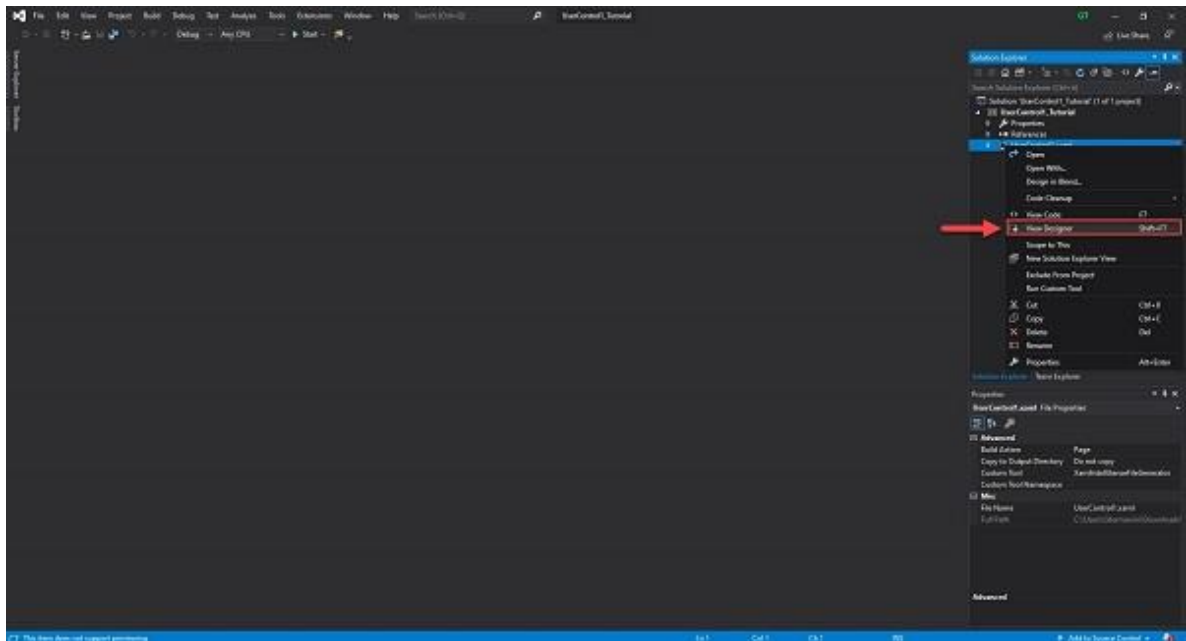


## Layout Definition

Once the project has been created, you will be able to start creating the control's graphics by opening the "UserControl1.xaml" file, which has been automatically created by the project wizard, by means of the 'View Designer' menu item (MAIUSC + F7).

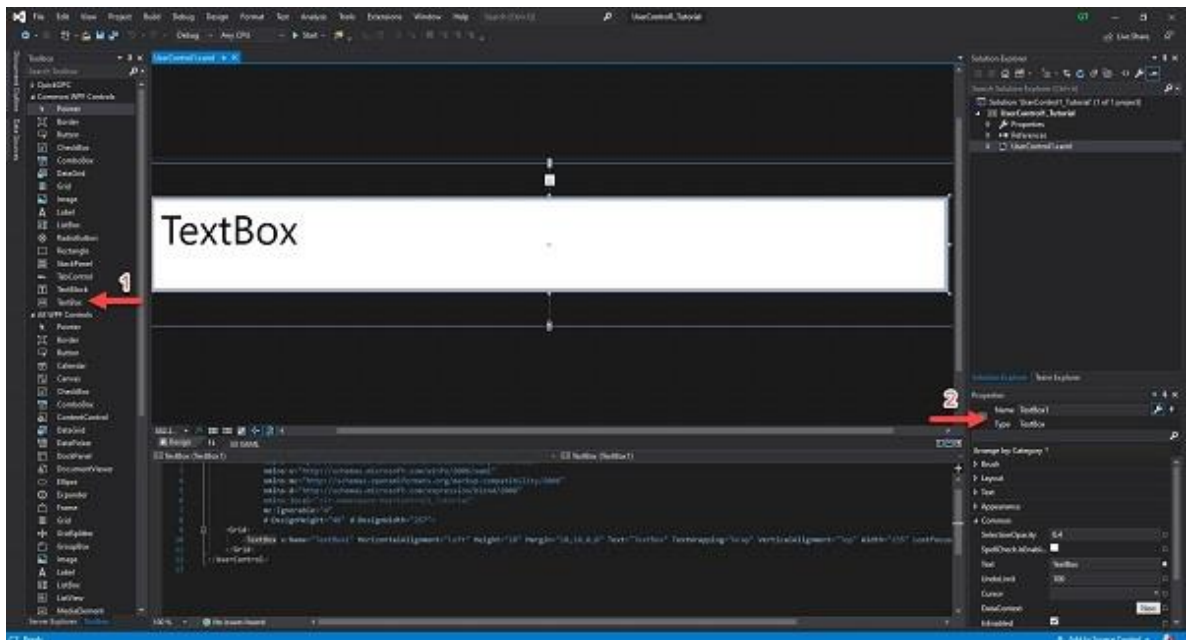


As an alternative, we can also edit its code directly it directly using the "View Code" menu item (Ctrl +Alt + E).

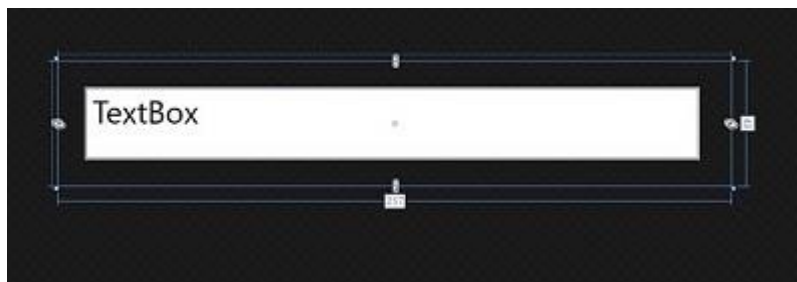


Now, insert a TextBox control by selecting it from the "Toolbox" menu, which is located for default on the left of the workspace, and through which we will display/set the Movicon.NEXT Tag.

By using the 'Property' pane, set the TextBox control's name as "TextBox1".

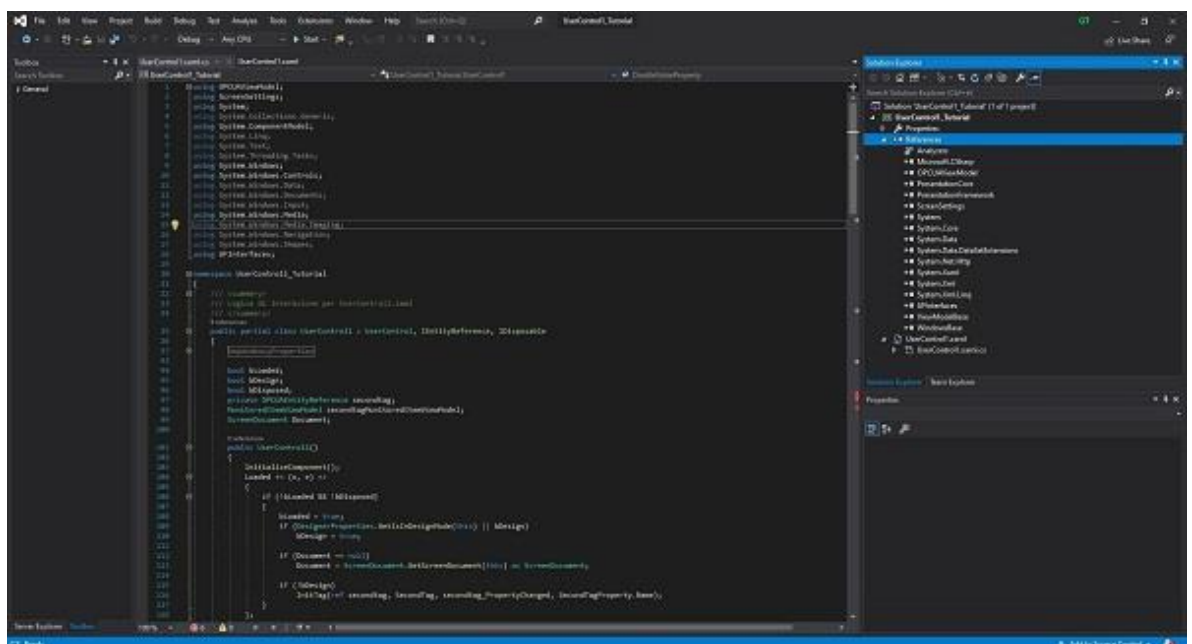
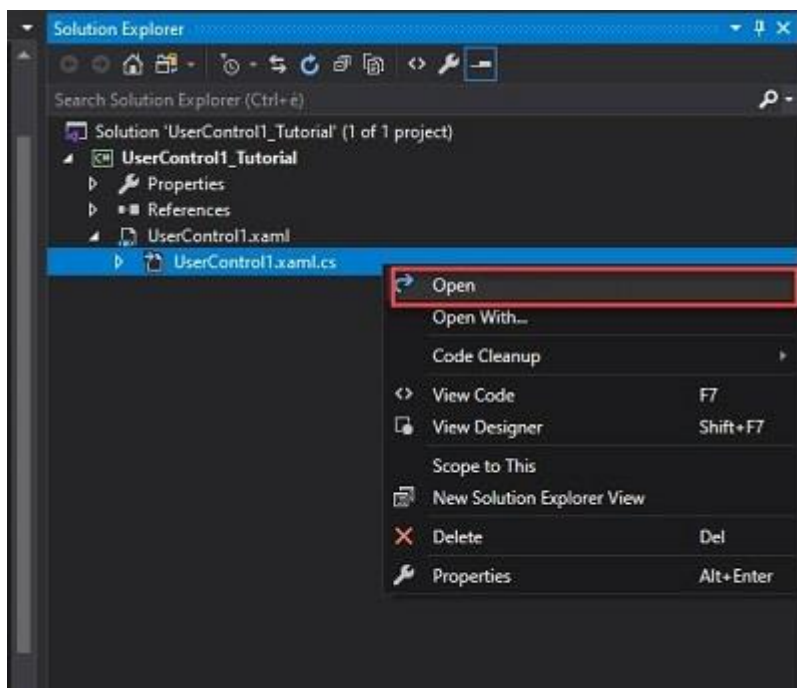


Finally, place the textbox correctly within the usercontrol and size it as follows:



## Implementing logic

Once the layout has been defined (in our case, built with the TextBox control) we must define the code to associate to the control. Therefore, open the Tab relating to the "UserControl1.xaml.cs" C# file:



A window will showing with our "UserControl1" UserControl base code:

```

9      using System.Windows.Documents;
10     using System.Windows.Input;
11     using System.Windows.Media;
12     using System.Windows.Media.Imaging;
13     using System.Windows.Navigation;
14     using System.Windows.Shapes;
15
16     namespace UserControl1_Tutorial
17     {
18         /// <summary>
19         /// Logica di interazione per UserControl1.xaml
20         /// </summary>
21         public partial class UserControl1 : UserControl
22         {

```

## Binding control properties and Tag

As you can see, even though the TextBox control is in the UserControl, it does not appear in the "UserControl1.xaml.cs" file's C# code as its definition resides in the corresponding associated "UserControl1.xaml" file. The WPF, in fact, keeps the application's layout elements, defined in XAML, and the specified control logic in C# separated.



We could have also defined and instantiated our TextBox control in C#, but we would not have the benefit of having the control's presentation separated from its logic.

We will now insert the code, which is needed to exchange data with Movicon.NExT, within the "UserControl1" class.

This we will do by using the WPF's binding feature to synchronize the TextBox control's value with a Movicon.NExT project's Tag value.

```

7      using System.Windows.Controls;
8      using System.Windows.Data;
9      using System.Windows.Documents;
10     using System.Windows.Input;
11     using System.Windows.Media;
12     using System.Windows.Media.Imaging;
13     using System.Windows.Navigation;
14     using System.Windows.Shapes;
15
16     namespace UserControl1_Tutorial
17     {
18         /// <summary>
19         /// Logica di interazione per UserControl1.xaml
20         /// </summary>
21         public partial class UserControl1 : UserControl
22         {
23
24             public UserControl1()
25             {
26                 InitializeComponent();
27             }
28         }
29     }

```

In this case, we shall insert the following code that specifies the "DoubleValue" Dependency Property before building it:

```

#region DependencyProperties
#region DoubleValue

```

```

    public static readonly DependencyProperty DoubleValueProperty =
DependencyProperty.Register("DoubleValue", typeof(double), typeof(UserControl1),
new    UIPropertyMetadata(0.0, new PropertyChangedCallback(OnDoubleValueChanged), new
CoerceValueCallback(OnCoerceDoubleValue)));
    private static object OnCoerceDoubleValue(DependencyObject o, object value)
    {
        UserControl1 control = o as UserControl1;
        if (control != null)
            return control.OnCoerceDoubleValue((double)value);
        else
            return value;
    }
    private static void OnDoubleValueChanged(DependencyObject o,
DependencyPropertyChangedEventArgs e)
    {
        UserControl1 control = o as UserControl1;
        if (control != null)
            control.OnDoubleValueChanged((double)e.OldValue, (double)e.NewValue);
    }
    protected virtual double OnCoerceDoubleValue(double value)
    {
        // TODO: Keep the proposed value within the desired range.
        return value;
    }
    protected virtual void OnDoubleValueChanged(double oldValue, double newValue)
    {
        // TODO: Add your property changed side-effects. Descendants can override as well.
        if (newValue != oldValue)
            TextBox1.Text = newValue.ToString();
    }
    public double DoubleValue
    {
        // IMPORTANT: To maintain parity between setting a property in XAML and procedural code, do not
touch the getter and setter inside this dependency property!
        get
        {
            return (double)GetValue(DoubleValueProperty);
        }
        set
        {
            SetValue(DoubleValueProperty, value);
        }
    }
#endregion
#endregion

```

```

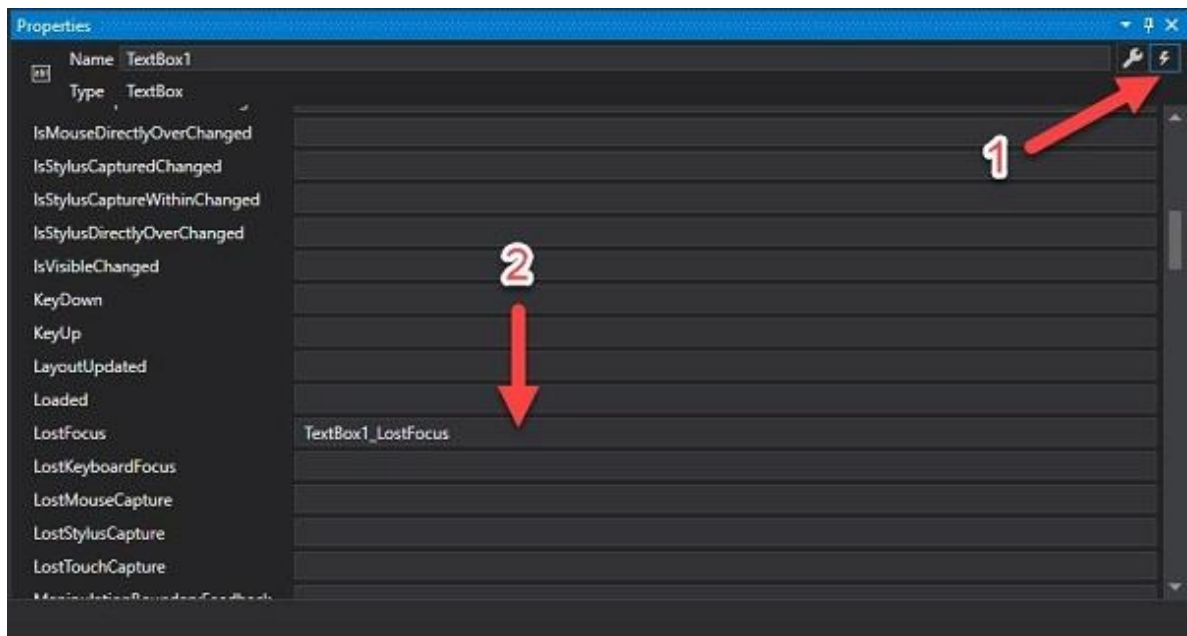
16 namespace MpfcControlLibrary
17 {
18     /// <summary>
19     /// Logica di interazione per UserControl1.xaml
20     /// </summary>
21     public partial class UserControl1 : UserControl
22     {
23         #region DependencyProperties
24
25         #region DoubleValue
26         public static readonly DependencyProperty DoubleValueProperty = DependencyProperty.Register("DoubleValue", typeof(double), typeof(UserControl1))
27
28         private static object OnCoerceDoubleValue(DependencyObject o, object value)
29         {
30             UserControl1 control = o as UserControl1;
31             if (control != null)
32                 return control.OnCoerceDoubleValue((double)value);
33             else
34                 return value;
35         }
36     }
37 }

```

By means of the "DoubleValue" property, any change of the Tag value in Movicon.NEXt will also change the Tag value in our TextBox object and viceversa.

We will now see how to update the property when TextBox control text changes.

Let's go back to the Graphic's design, select the UserControl1.xaml file and select the event upon which the text contained in the TextBox will be sent to Movicon: in our case, we will use the "LostFocus" event. Double click on this event:



This will take us to the part where we develop the code and where Visual Studio will have created the event's definition that we previously chose on the list.

We will insert the code needed within the definition to set the property's value based on the TextBox's text by making sure that it is numeric type (double):

```

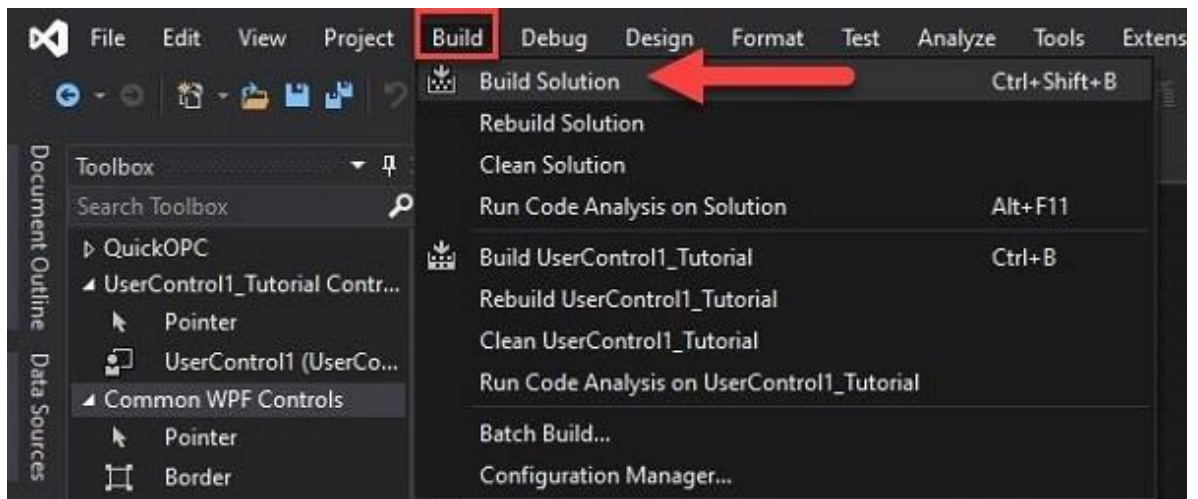
try
{
    DoubleValue = Double.Parse(TextBox1.Text);
}
catch (Exception)
{
    TextBox1.Text = DoubleValue.ToString();
}

```

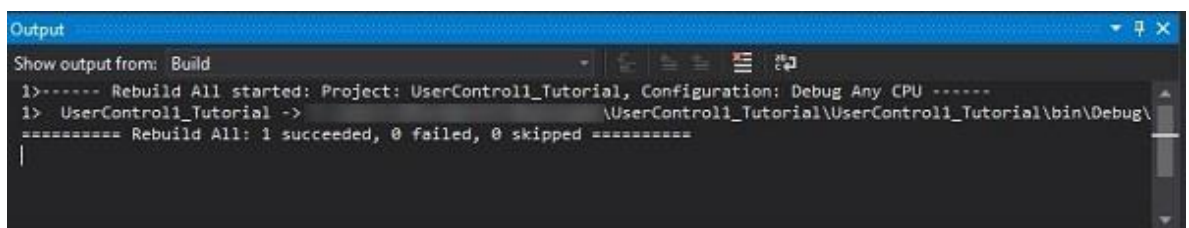
## Build and deploy new control

Select the 'Build solution' item from the 'Build' menu.





Once built, we should see a message, like the one below, in the output window:



The compilation generates a file with .dll extension which should be copied to the Movicon.NExT installation folder's Toolbox subfolder, like so:

**"C:\Program Files\Progea\Movicon.NExT x.x\Toolbox"**

Once the dll file has been copied, we shall create the link to our library within the Movicon.NExT Toolbox. Now open this path:

**"C:\ProgramData\Progea\Movicon.NExT. x.x\Toolbox\Screen"**

Here, we will see a series of folders, each one corresponding to a group in the Movicon.NExT Toolbox. We shall create a new folder, e.g. "MY\_CONTROLS", which will be the name of the group in the Movicon.NExT Toolbox..

Once we have created the folder, we will enter and insert the files of our library inside.

We now have to create 2 files:

- "TestControllo1.png" which will be the icon that we will display in the Toolbox that corresponds to our control (for the time being, we shall copy the one presented as "Displays/EditDisplay.png" and rename it as above);
- "TestControllo1.xaml" which will be the library description file.

The "TestControllo1.xaml" file contains the information needed by Movicon.NExT to call the library we have previously created.

## Detailed Analysis of the XAML file for the Toolbox

In our example, the "TestControllo1.xaml" file will contain:

```
<tr:UserControl1 xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Tag="ProblematicXamlWriter"
  xmlns:tr="clr-namespace:UserControl1_Tutorial;assembly=UserControl1_Tutorial"
  Width="250" Height="60" Background="{x:Null}"
```

```
xmlns:mynsp="clr-namespace:Converters;assembly=Converters">
```

```
<tr:UserControl1.Resources>
```

```
<!-- to force interoperability on webclient -->
```

```
<Style TargetType="{x:Type ListView}" x:Key="ConnectionStringBind" />
```

```
<!-- to use our stringtodouble converter -->
```

```
<mynsp:UpDownValueConverter x:Key="ValueConverter" />
```

```
</tr:UserControl1.Resources>
```

```
<tr:UserControl1.DoubleValue>
```

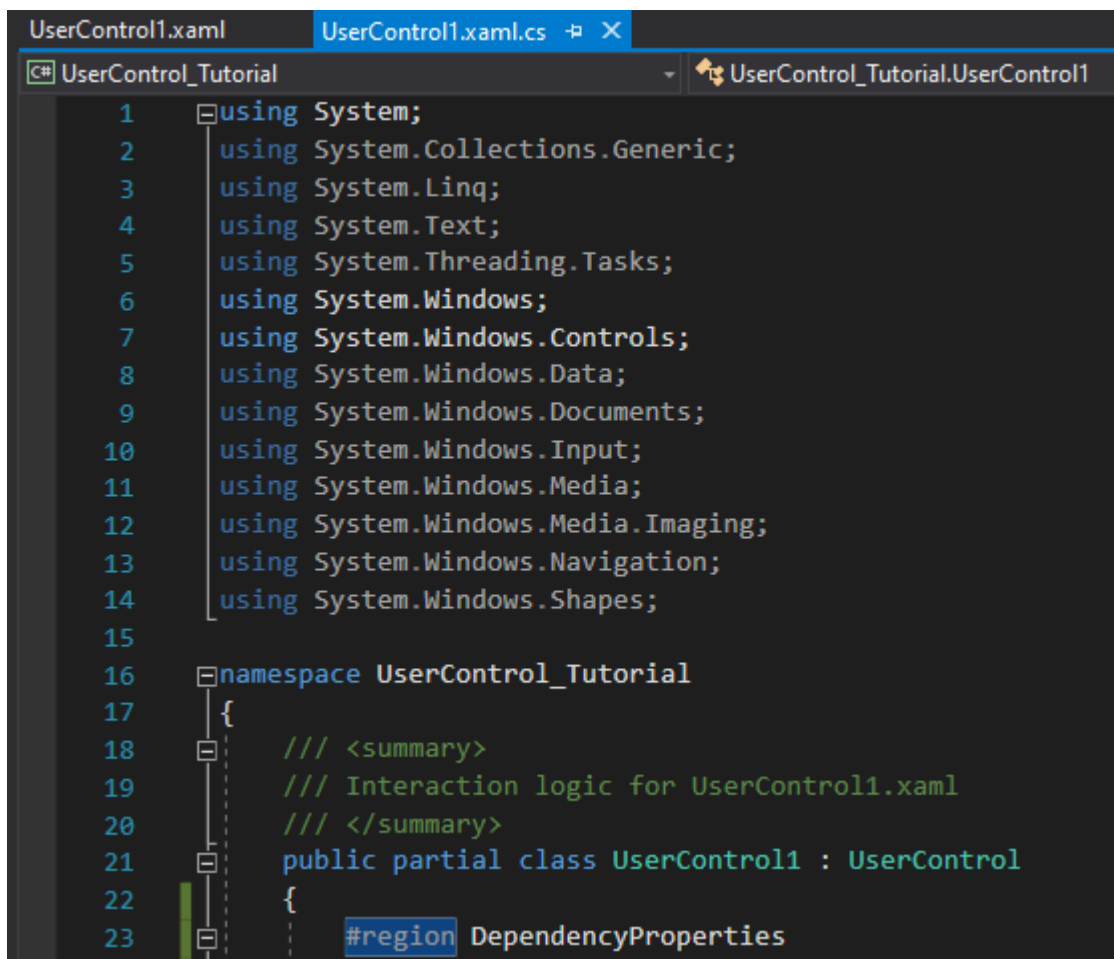
```
<Binding Path="Value" Converter="{StaticResource ValueConverter}" Mode="TwoWay"
ValidatesOnDataErrors="True" UpdateSourceTrigger="PropertyChanged"></Binding>
```

```
</tr:UserControl1.DoubleValue>
```

```
</tr:UserControl1>
```

**xmlns:tr="clr-namespace:UserControl\_Tutorial;assembly=UserControl\_Tutorial"**

It will contain the name space of our library which we can retrieve from the Visual Studio here:



**Width="250" Height="60"**

This consents the width and height default values of the User Control inserted on screen to be defined.

**xmlns:mynsp="clr-namespace:Converters;assembly=Converters">**

This consents the use of the Converter which has already been defined within Movicon.NExT to transform data between different types.



**Tag="ProblematicXamlWriter"**

**<Style TargetType="{x:Type ListView}" x:Key="ConnectionStringBind" />**

These are the instructions needed for the correct serialization of the control and the handling of any eventual connection strings to the database, used internally by Movicon.NExT.

**<tr:UserControl1.Resources> e <tr:UserControl1.DoubleValue>**

This consents a read me file to be created between the control's variable and Movicon.NExT's variable.

**<mynsp:UpDownValueConverter x:Key="ValueConverter" />**

In this example, the two-way converter is used to convert data from string to double data.



Note carefully the Binding instruction:

**<Binding Path="Value" Converter="{StaticResource ValueConverter}"  
Mode="TwoWay" ValidatesOnDataErrors="True"  
UpdateSourceTrigger="PropertyChanged"> </Binding>**

This instruction establishes the connection between the "DoubleValue" property's value of our UserControl and the value of the tag associated to the object from within Movicon.NExT.

Now, let's look at each individual attribute:

**Path="Value"**

This attribute is not to be changed, as the "value" property represents the value of the Movicon Tag value associated to the "DoubleValue" property defined by us in the UserControl.

**Converter="{StaticResource ValueConverter}"**

Even though the "DoubleValue" property has been defined by us as a double type, the Tag's value is handled internally as a string. Therefore, it will be our job to use the converter, inserted in this case as a static resource, for eventual conversions.

**Mode="TwoWay"**

The two-way binding mode which keeps our UserControl's DoubleValue property synchronized with the Movicon Tag value. The variation of one of these will automatically update the other one likewise.

**ValidatesOnDataErrors="True"**

This instruction consents any error messages to be displayed for the user to see in cases where the parent object implements the IDataErrorInfo interface upon variation of the bound value.

**UpdateSourceTrigger="PropertyChanged"**

This attribute specifies the binding's activation logic to update the property (in this case, in correspondence to the PropertyChanged event).

## Insert and test UserControl

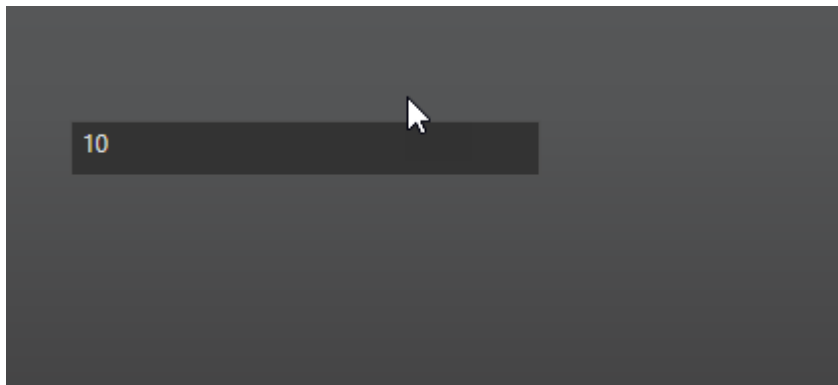
Once the file has been saved and Movicon.NExT has been launched, we will find our control among those available in the ToolBox, in the "MY\_CONTROLS" folder (the name we chose previously).

Now add the control we have just created to a screen and associate it with a Double type tag by dragging it from the project explorer window onto the control. To make things easy, we will assign the tag with an initial value of 10.

Once this operation has concluded, we can check to see if the Tag has been assigned correctly by opening the control's Tag property from the property pane.

We will now startup our project in runtime. If we have done everything correctly we should see that:

1. TextBox of our UserControl shows the Tag's initial value 10, thanks to binding with the "DoubleValue":



2. As the TextBox control is editable, we can specify a value to assign to the associated Tag.

## Further details

### Distinction between Design and Runtime

For the UserControls to function correctly and to improve their performances, it always best that their behaviour is managed according to where they are used in development or runtime mode.

In particular, it is always best to insert the code below in the builder or methods inserted in the UserControl code (C#) which must discriminate its behaviour::

```
if (DesignerProperties.GetIsInDesignMode(this) || bDesign)
{
    bDesign = true;
}
else
{
}
```

In addition, it would be best to discriminate the operations to be performed only when first loaded, in respect to those which can be called after (for example for exposed property variations).

Eg.:

```
//Constructor
bool bLoaded;
public UserControl1()
{
    Loaded += (o, e) =>
    {
        if (!bLoaded)
        {
            bLoaded = true;
            if (DesignerProperties.GetIsInDesignMode(this) || bDesign)
            {
                bDesign = true;
            }
            else
            {
            }
        }
    };
}
```

### Data Binding to Tag by code

In cases where you wish to free the control from the DataBinding of a specific property in the ToolBox's object's XAML, it is possible to manage the binding to the control using the DataContextChanged event in this way:

```
DataContextChanged += (o, e) =>
{
    if (bDispose)
        return;

    if (DataContext is OPCUAViewModel.MonitoredItemViewModel &&
        !DesignerProperties.GetIsInDesignMode(this))
    {
        if (monitoredItemViewModel != null)
            monitoredItemViewModel.PropertyChanged -=
monitoredItemViewModel_PropertyChanged;

        monitoredItemViewModel = DataContext as
OPCUAViewModel.MonitoredItemViewModel;

        if (monitoredItemViewModel == null)
            return;
        try
        {
            {
                DoubleValue =
Double.Parse(monitoredItemViewModel.DataValue.Value.ToString());
            }
            catch (Exception)
            {
            }
            UpdateDisplay();
            monitoredItemViewModel.PropertyChanged +=
monitoredItemViewModel_PropertyChanged;
        }
    }
};
```

### Memory management: Dispose

In the specific management via DataContextChanged event example, and in all the other cases in which the control necessitates hoc logic to free used memory, it would be best to implement the IDisposable interface as follows:

```

public void Dispose()
{
    if (!bDispose)
        bDispose = true;
    /* insert here disposing code*/

    if (monitoredItemViewModel != null)
        monitoredItemViewModel.PropertyChanged -=
            monitoredItemViewModel_PropertyChanged;

    monitoredItemViewModel = null;
}

```

### Custom Editor for complex properties

It may be handy to provide a custom editor to valorize simple or complex properties defined within the UserControl, such as a slider that allows its numeric values to be edited.

In cases like this, if the UserControl implements a special interface, when the user is about to edit the control's properties in design mode, Movicon.NEXt will automatically integrate and load the editor specified in the property panel.

To make this happen, implement the "IContainPropertyEditors" interface by defining the methods and properties as follows:

```

#region IContainPropertyEditors Members

[Browsable(false)]
public Type ObjectType
{
    get
    {
        return this.GetType();
    }
}

[Browsable(false)]
public IDictionary<string, DataTemplate> GetListDataTemplates
{
    get
    {
        var mapDataTemplates = new Dictionary<string, DataTemplate>();

        // Defines Data Template for 'Min/MaxValueProperty' dependency property.
        var dt = new DataTemplate();
        var factory = new FrameworkElementFactory(typeof(MyPropertyEditor));
        dt.DataType = typeof(double);
        dt.VisualTree = factory;
        mapDataTemplates.Add(MinValueProperty.Name, dt);
        mapDataTemplates.Add(MaxValueProperty.Name, dt);

        return mapDataTemplates;
    }
}

#endregion

```

### SmartControl

The above described solution may not be suitable for some UserControls. One example would be editing pens of Trend control types where it would result necessary to develop complex controls that can be configured within a dedicated space to avoid overcrowding the properties window.

In order to do this, there is a special "SmartControl" property that has to be defined within the UserControl as follows:

```

UserControl smartControl;
[Browsable(false)]
public UserControl SmartControl

{
    // IMPORTANT: To maintain parity between setting a property in XAML and
    // procedural code, do not touch the getter and setter inside this dependency property!
    get
    {
        if (smartControl == null)
            smartControl = new MySmartControl(this);
        return smartControl;
    }
}

```

Where "MySmartControl" represents the UserControl that it should implement for managing the control's specific properties.

### Definition of other properties connected to Tags

We have seen in the section on binding how it is possible to synchronize the value of the Tag associated to the control with the value of the UserControl's property.

This is also valid for contextual Tags which are those associated to the object using the 'Tag' property made available by default (it can also be set by dragging it from the I/O Data Server).

However, it still remains possible to define additional UserControl properties to connect project Tags ; This is already done, for example, for the "EditDisplay" control with the "Tag Min Value" and "Tag Max Value" properties.

To do this, you will need to define a new "OPCUAXMLEntityReference" Dependency type property which we will call "SecondTag" to simplify things:

```

#region SecondTag
public static readonly DependencyProperty SecondTagProperty = DependencyProperty.Register("SecondTag",
    typeof(OPCUAXMLEntityReference), typeof(UserControl1), new UIPropertyMetadata(null));

public OPCUAXMLEntityReference SecondTag
{
    get
    {
        return (OPCUAXMLEntityReference)GetValue(SecondTagProperty);
    }
    set
    {
        SetValue(SecondTagProperty, value);
    }
}
}
#endregion

```

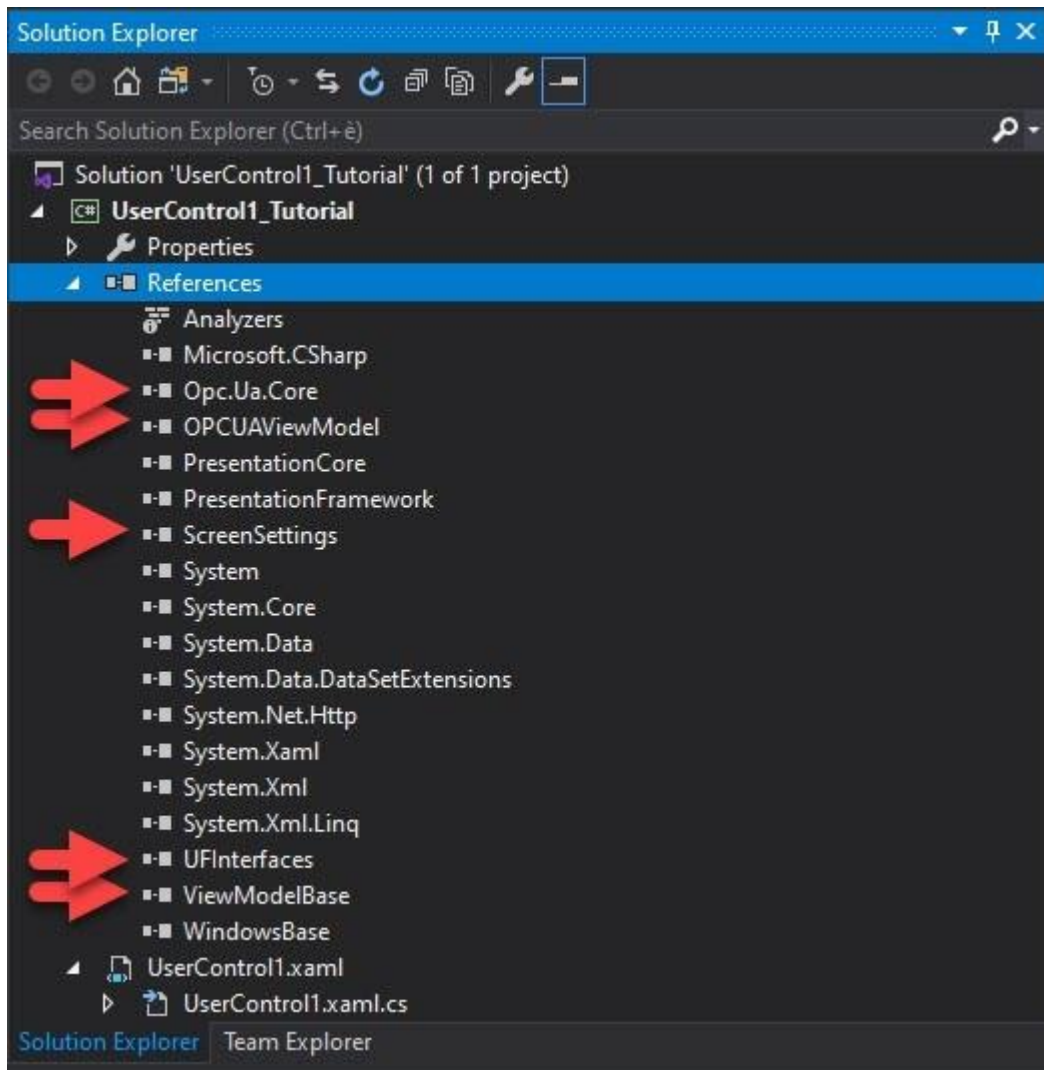
### Add missing references

After having inserted the property, the application will remit an error while compiling it, this is because the "OPCUAXMLEntityReference" type belongs to a Movicon.NExT assembly that is not referenced by this project.

In order to proceed in implementing it and thus compile the DLL of our UserControl, we will need to reference the following assemblies in the Movicon.NExT installation folder:

1. OPCUAViewModel
2. Opc.Ua.Core

3. ScreenSettings
4. UFIInterfaces
5. ViewModelBase



### Implementing interfaces

Once done, we now have to implement our UserControl with the "IEntityReference" and "IDisposable" interfaces:

```

19
20 namespace UserControl1_Tutorial
21 {
22     /// <summary>
23     /// Logica di interazione per UserControl1.xaml
24     /// </summary>
25     public partial class UserControl1 : UserControl, IEntityReference, IDisposable
26     {
27         #region DependencyProperties
28
29         DoubleValue
30
31         #region SecondTag

```

[...]

```

#region IEntityReference
[Browsable(false)]
public ImageSource CollapsedImageSource => null;

[Browsable(false)]
public ImageSource ExpandedImageSource => null;

[Browsable(false)]
public ContextMenu contextMenu => null;

[Browsable(false)]
public object Tooltip => null;

[Browsable(false)]
public object ContainedObject => null;

[Browsable(false)]
public object EntityParent => null;

[Browsable(false)]
public string TypeDefinitionString => null;
#endregion

```

[...]

```

#region IDisposable
public void Dispose()
{
    if (bDisposed)
        return;
    bDisposed = true;
}
#endregion

```

### Further implementations

We will now continue with updating the builder of our UserControl to initialize the Tag connected to the new property:

```

public UserControl1()
{
    InitializeComponent();
    Loaded += (o, e) =>
    {
        if (!bLoaded && !bDisposed)
        {
            bLoaded = true;
            if (DesignerProperties.GetIsInDesignMode(this) || bDesign)
                bDesign = true;

            if (Document == null)
                Document = ScreenDocument.GetScreenDocument(this) as ScreenDocument;

            if (!bDesign)
                InitTag(ref secondtag, SecondTag, secondtag_PropertyChanged, SecondTagProperty.Name);
        }
    };
}

private void InitTag(ref OPCUAEntityReference preparedtag, OPCUAXMLEntityReference xmltag,
    PropertyChangedEventHandler propertyChangedEventHandler, string property)
{
    if (preparedtag == null && xmltag != null && xmltag.TagReference != null)
        preparedtag = xmltag.TagReference;
    if (preparedtag != null && !preparedtag.IsRelative)
        PrepareExecution(nameof(UserControl1), Document, this, propertyChangedEventHandler, preparedtag);
}

```

This is followed by the two “PrepareExecution” and “TerminateExecution” methods which are indispensable for resolving and putting the Tag referenced by the UserControl in runtime:

```

void PrepareExecution(String SessionName, ScreenDocument Document, IEntityReference er,
    PropertyChangedEventHandler propertyChangedEventHandler, OPCUAEntityReference tag)
{
    if (tag != null && tag.IsValid)
    {
        tag.PropertyChanged -= propertyChangedEventHandler;
        if (bDisposed)
            return;

        tag.PropertyChanged += propertyChangedEventHandler;

        if (Document != null && !string.IsNullOrEmpty(Document.SessionString))
            tag.Resolve(Document.SessionString, Document);
        else
            tag.Resolve(SessionName, Document);

        tag.SetInUse(er, true);
    }
}

public void TerminateExecution(IEntityReference er, PropertyChangedEventHandler propertyChangedEventHandler,
    PropertyChangedEventHandler monitoredPropertyChangedEventHandler, OPCUAEntityReference tag, MonitoredItemViewModel monitored)
{
    if (monitored != null)
        monitored.PropertyChanged -= monitoredPropertyChangedEventHandler;

    if (tag != null)
    {
        tag.PropertyChanged -= propertyChangedEventHandler;

        if (tag.IsValid)
            tag.SetInUse(er, false);
    }
}

```

The following are then implemented:

1. The 'EventHandler which will be executed upon changing the Tag's reference (secondtag\_PropertyChanged)
2. The method that processes and recalls the logic needed to change the Tag's value (secondtagMonitoredItemViewModel\_PropertyChanged)



```

private void _secondtag_PropertyChanged(object sender, PropertyChangedEventArgs e)
{
    OPCUAEntityReference n = (OPCUAEntityReference)sender;
    if (e.PropertyName == "MonitoredItemViewModel")
    {
        if (_secondtagMonitoredItemViewModel != null)
            _secondtagMonitoredItemViewModel.PropertyChanged -= _secondtagMonitoredItemViewModel_PropertyChanged;

        if (bDisposed)
            return;

        _secondtagMonitoredItemViewModel = n.MonitoredItemViewModel;

        if (_secondtagMonitoredItemViewModel != null)
        {
            _secondtagMonitoredItemViewModel.PropertyChanged += _secondtagMonitoredItemViewModel_PropertyChanged;

            _secondtagMonitoredItemViewModel_PropertyChanged(
                _secondtagMonitoredItemViewModel, new PropertyChangedEventArgs("DataValue")
            );
        }
    }
}

```

```

private void _secondtagMonitoredItemViewModel_PropertyChanged(object sender, PropertyChangedEventArgs e)
{
    if (bDisposed)
        return;

    MonitoredItemViewModel m = (MonitoredItemViewModel)sender;
    if (m == null)
        return;

    if (e.PropertyName == "LastMessage")
    {
        //SetError(m.LastMessage);
    }
    else if (e.PropertyName == "DataValue")
    {
        if (m.DataValue != null)
        {
            if (Opc.Ua.StatusCode.IsGood(m.DataValue.StatusCode) ||
                m.DataValue.StatusCode == Opc.Ua.StatusCodes.UncertainLastUsableValue)
            {
                try
                {
                    if (m.DataValue.Value != null)
                    {
                        //... DataValue.Value value handling
                        //MessageBox.Show("VALORE NUOVO: " + m.DataValue.Value.ToString());
                    }
                }
                catch (Exception)
                {
                }
                //UnsetError();
            }
            //else
            //SetError(m.DataValue.StatusCode.ToString());
        }
    }
}

```

As you see above, the new tag value is represented by the "DataValue.Value" property of the "MonitoredItemViewModel" object type.

Upon reception of the new value, the logic should be implemented and which is needed to update the status of the UserControl as a consequence (commented example).

Instead, the "DataValue.LastMessage" property indicates the presence of an error message (it is the UserControl's developer's job to manage this).

Finally, implementation of the "Dispose()" method is shown below and it is the method which invokes the "TerminateExecution" to unsubscribe from notifications relating to the Tag:

```
public void Dispose()
{
    if (bDisposed)
        return;
    bDisposed = true;

    TerminateExecution(this, secondtag_PropertyChanged,
        secondtagMonitoredItemViewModel_PropertyChanged, secondtag, secondtagMonitoredItemViewModel);
}
```

## Attachments

### Visual Studio Solution

In the file .zip distributed together with this document you will find an attachment containing the example project described above.



In order to compile it, you will need to add the references to the DLL indicated at point 7.5.1, residing in the Movicon installation folder, when indicated as an compilation error, and update the project's "Target Framework" to the .NET Framework version as required by these assemblies.

### Movicon.NExT Toolbox Files

The files to be copied in the Toolbox folder under ProgramData have been put back in the "Toolbox" Folder as indicated above at point "Build and deploy new control" .

## 6.3. Assigning Tags to objects

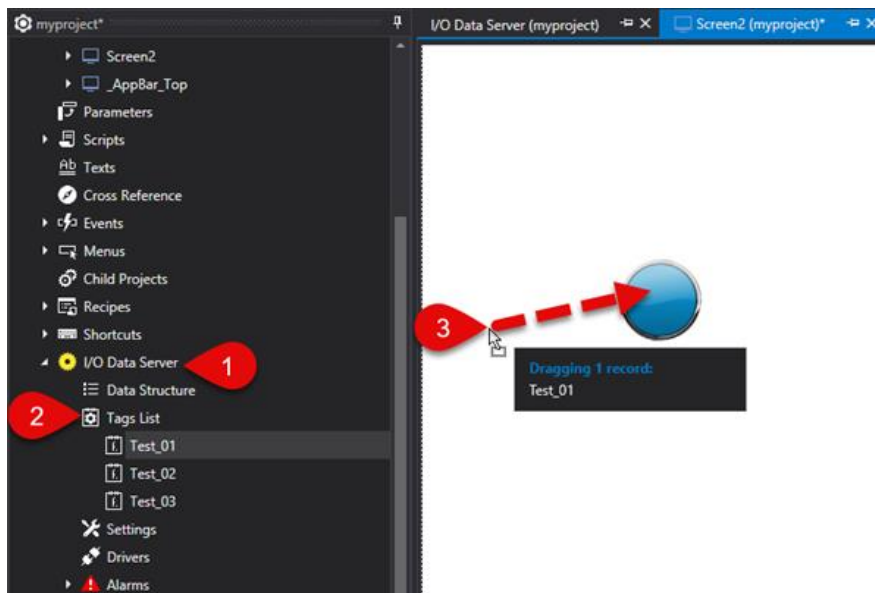
Each graphical element (objects from the Toolbox or Symbols from the Symbol Library) can be associated with a Tag. This Tag will be considered as the reference Tag for each associated function (such as Commands and Animations) when not specified otherwise with a specific Tag for a specific function of the object.

## Assigning Techniques

Assigning the Tag to a graphic object can be done using different techniques as described below.

### Assigning tags with Drag and Drop

Tags can be assigned to object by simply dragging the Tag to the actual object using the Drag&Drop techniques.



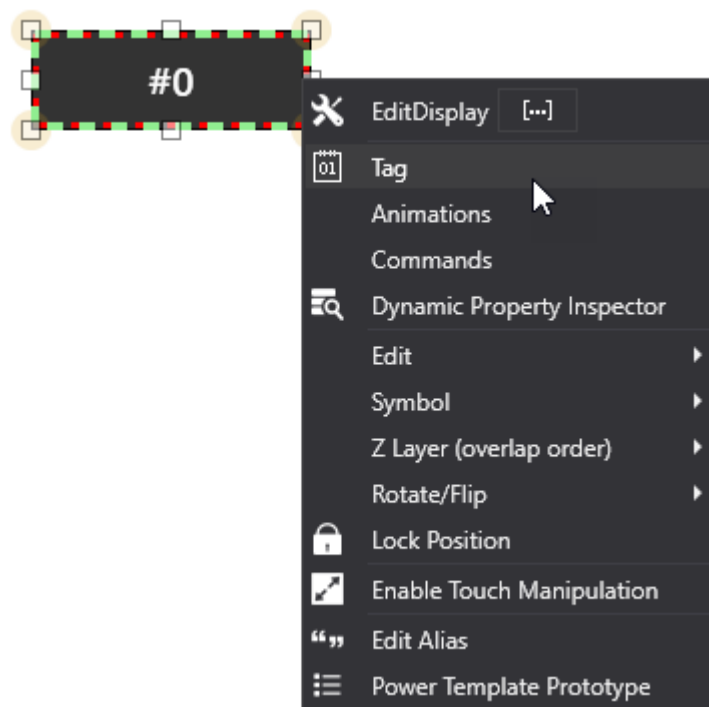
This procedure is as follows:

1. Expand the I/O Data Server resource in the Project's Explorer window
2. Expand the Tag list
3. Select the desired Tag from the list and drag it on to the desired graphical object on screen.

This will automatically assign the Tag to the object after which it will be possible to set its properties.

### Assigning Tags using contextual menu

Tags can be assigned to objects by calling the contextual menu and selecting the Tag item. This will open the Tag Explorer Window from which to select the Tag.



### Assigning Tags using the Properties Window

Tags can be assigned to objects by calling the property window and selecting the Tag item. A Tag Explorer Window will open to select the Tag.

### Assigning Tags using rapid commands

Tags can be assigned to objects by pressing down the SHIFT key while double clicking on the object. This will open a Tag Explorer Window to select the Tag.

## TAG Explorer Window

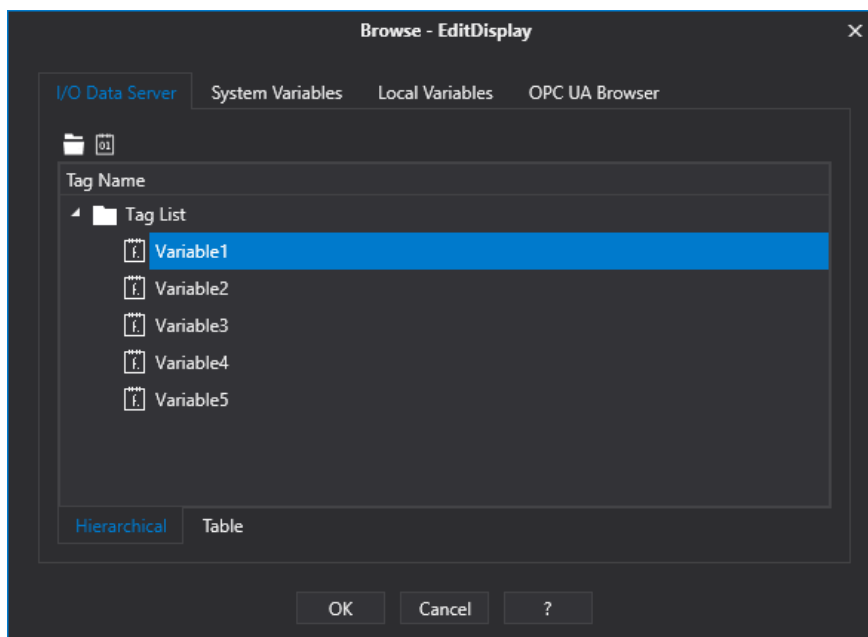
Once opened, the "Tag Explorer Window" presents four tabs for selecting the tag to associate to the object.

Each tab presents a different variable typology:

- I/O Data Server
- System Variables
- Local Variables
- OPC UA Browser

### I/O Data Server

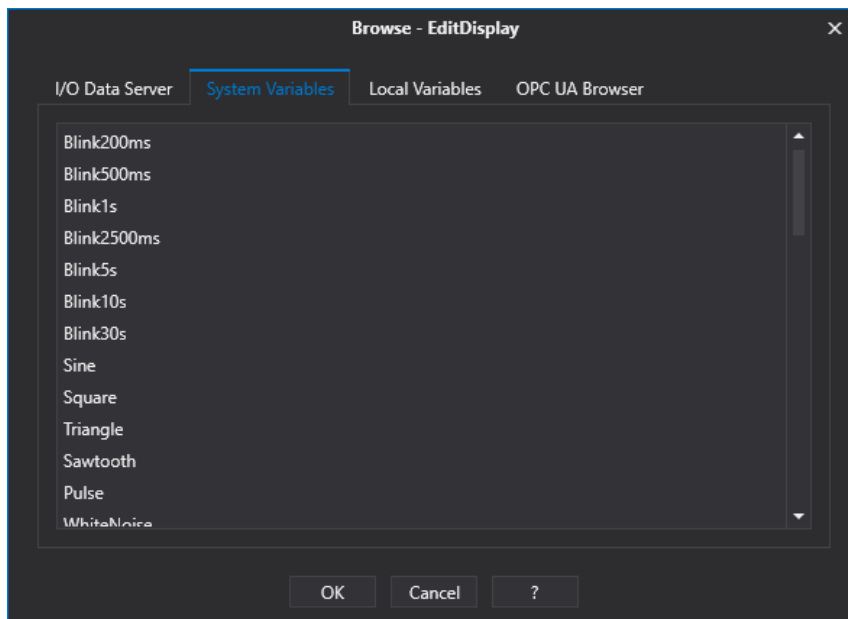
The first tab reports the list of Tags defined in the project's I/O Data Server.



In this context it is also possible to perform Tag editing operations, whereby new tags or folders can be added, changed, deleted or already existing tags can be moved.

### System Variables

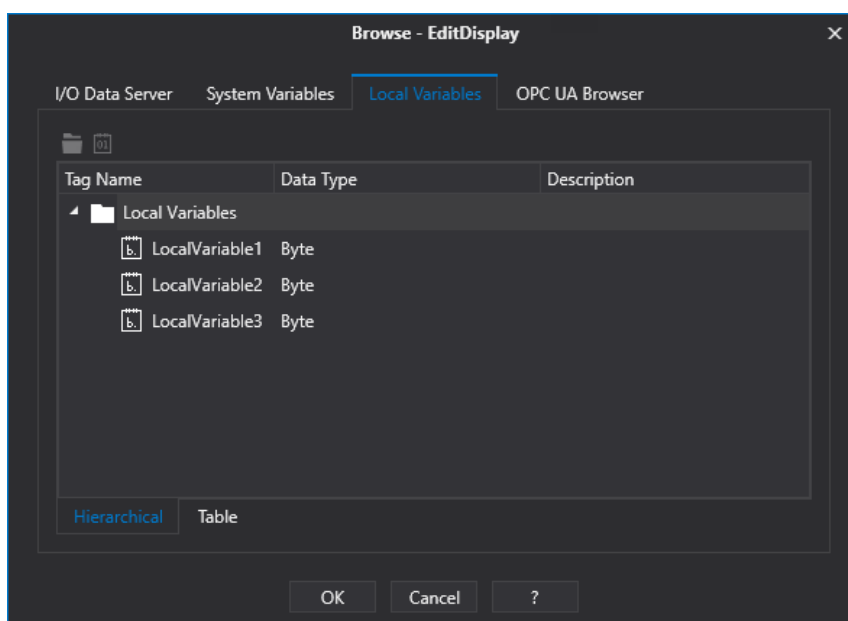
The second tab reports a list of the Client's "System Variables" that allow you to obtain system information or simulation curves (see paragraph: "Client System Variables").



### Local Variables

The third tab reports the "Local Client variables" which are those variables that have been defined at Client level.

(see paragraph "Local Client Variables").

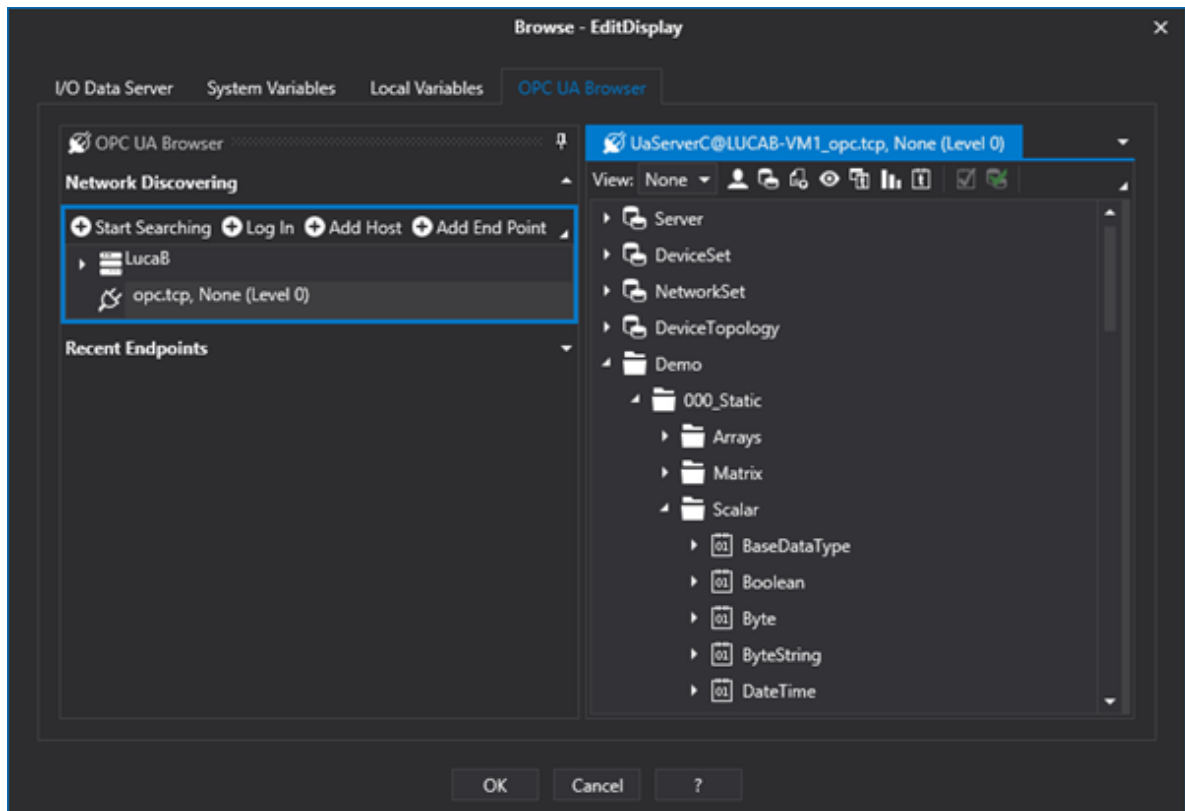


In this context it is also possible to perform Tag editing operations, whereby new tags or folders can be added, changed, deleted or already existing tags can be moved.

### OPC UA Browser

The third tab called "OPC UA Browser" allows you to go and select a tag exposed by OPC UA Servers that are present in network.

For further information on how to used the "OPC UA Browser" commands please refer to the paragraph on "OPC UA Browser Window" that used the same functionality.



As the Movicon.NExT I/O Data Server is a OPC UA Server, the OPC UA Browser can also be used to select variables from the project's I/O Data Server which should be running.

## 6.4. Use of Expressions in objects

### 6.4.1. Pointing to Variable Bits and Array Variable Elements

Movicon.NExT supports the possibility to specify and assign a single bit of a byte, word, or double variable type or a single element of a data Array when assigning Tag variables to symbols or objects.

#### Pointing to bits

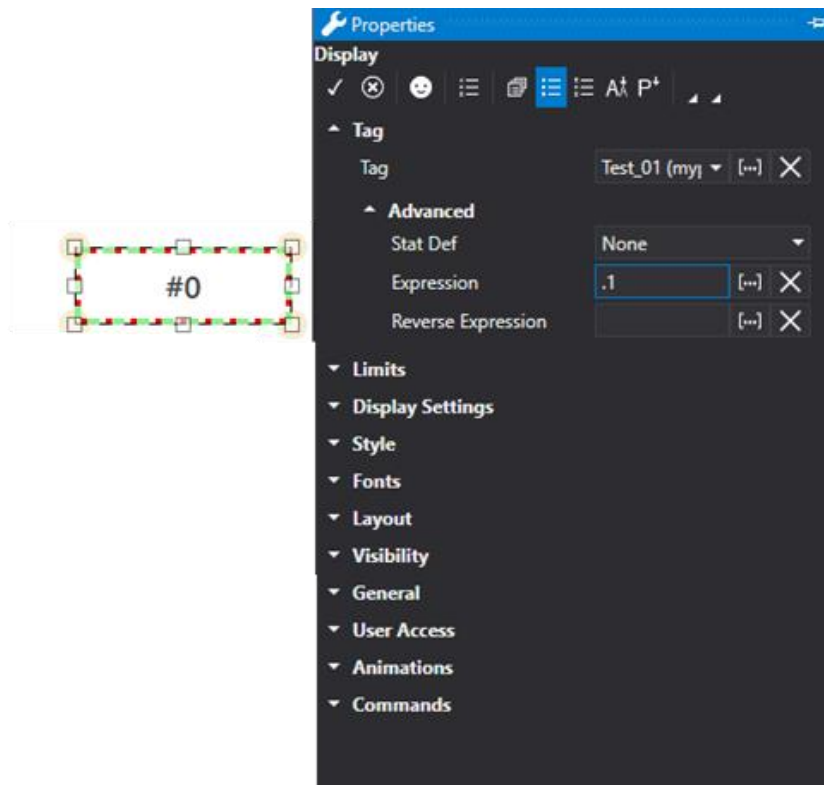
To specify the bit of a Tag variable assigned to a symbol or an object, you will need to use the **"Input Expression"** field which can be accessed using the right mouse key or the actual object's command menu.



The bit numbers (starting with 0) to be assigned must be entered in the **expression** field after a dot. For example bit 1 should be inserted as ".1".



The Inverse or Reverse Expressions are not necessary in these cases.



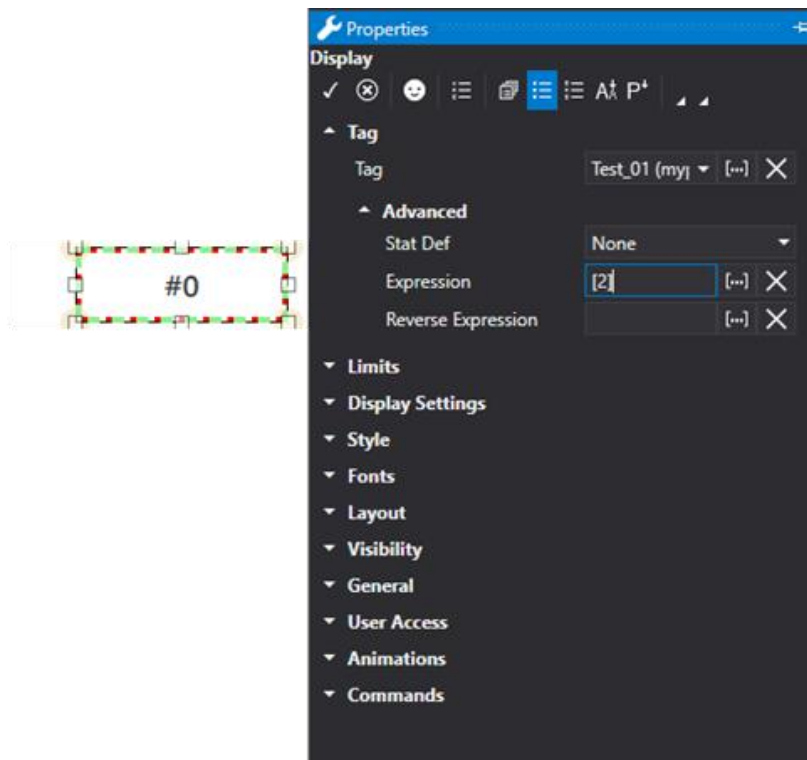
In order to use the bits within the logic functions you will need to use the BITAND function for example:

- **=AND(BITAND([x],4), BITAND([x],16))**  
This expression executes a AND logic between the bit 2 and the bit 4 of the variable [x] (because  $2^2 = 4$  and  $2^4 = 16$ ).  
This type of expression is to be used in substitution of a "=AND([x].2, [x].4)" expression type which is not allowed.
- **=OR(BITAND([x],8), BITAND([x],16))**  
This expression executes an OR logic between the bit 3 (because  $2^3 = 8$ ) and the bit 4 of the variable [x] (because  $2^4 = 16$ ).  
This type of expression is to be used in substitution of a "=AND([x].3, [x].4)" express type which is not allowed.  
The "[x]" placeholder represents the value of the Tag associated to the object's Tag property.

## Pointing to Array elements

An element from an **Array Type Tag** variable can be assigned in the same way as pointing to a variable bit. Also in this case, after having associated the Array variable to the object, the number of the element to point to must be inserted in the ""Input Expression Code" field by specifying it between square brackets.

For example, to point to the first Array element, in other words element 0, you will need to insert the "[0]" syntax. .



Array elements can also be used within logic functions and other functions such as:

- **= AND([x(2)], [2:Tags/2:myBoolArr(3)])**  
This expression executes an AND logic between element 2 of the "[x]" array with element 3 of the "2:Tags/2:myBoolArr" array.
- **= AND([x(2)], [x(3)])**  
This expression executes an AND logic between element 2 of the "[x]" array with the element 3 of the "[x]" array. The "[x]" placeholder represents the value of the Tag associated to the object's Tag property.

### 6.4.2. Other Expressions

It is possible to insert expressions or formulae in the object's 'Expressions' and 'Inverse Expressions' property fields (or Animation properties).

This possibility enables the system to calculate the set expression and display the result as an assigned object value.

#### Expressions in objects

If you wish to the value in the display resulting from an expression between a Tag and a constant calculation, you will need to assign the Tag to the object as a Contextual Tag and then insert the calculation expression in the Display's 'Expression' property. From this point onwards, the Display will no longer display the Tag value but the resulting expression's value at runtime.



**IMPORTANT:** when inserting an Expression in an object, you will need to assign one of the Tags managed in the expression as the object's contextual Tag.

In cases in which the object, for example the Display object, should be editable, it will be necessary to insert the 'Inverse Expression' by means of which the value's bidirectionality can be managed. When the



control is also editable, it is necessary to retrieve the value with which to set the variable associated to the object when the control is edited.

For example, associating the "TagName1" variable to a Display object and then inserting the following formula in the "Expression" field:

$$=[x] + [TagName2] + 10$$

Where the placeholder "[x]" represents the value of the contextual Tag associated to the control. The square brackets, however, are used to indicate that the content is a Movicon.NExT variable. .

At this point, if the dynamic value of the "TagName1" associated to the Display object is 1, and the value of the "TagName2" is 2, the displayed value will be the result of the expression, that is to say:

$$1 + 2 + 10 = 13$$

If the Display were also editable, it would be necessary to apply the inverse formula to set the variable according to the inserted value. In this case, the 'Inverse Expression' field would be edited with the following formula:

$$=[x] - [TagName2] - 10$$

Where, in this case, the placeholder [x] represents the value inserted in the display object (and not the value of the Contextual Tag) and the result of the formula is then inserted in the contextual Tag associated to the object, for example "TagName1".

At this point, if the value inserted in the Display 20 and the value of the 'TagName' were 2, the 'TagName1' would be set with the resulting value of the expression:

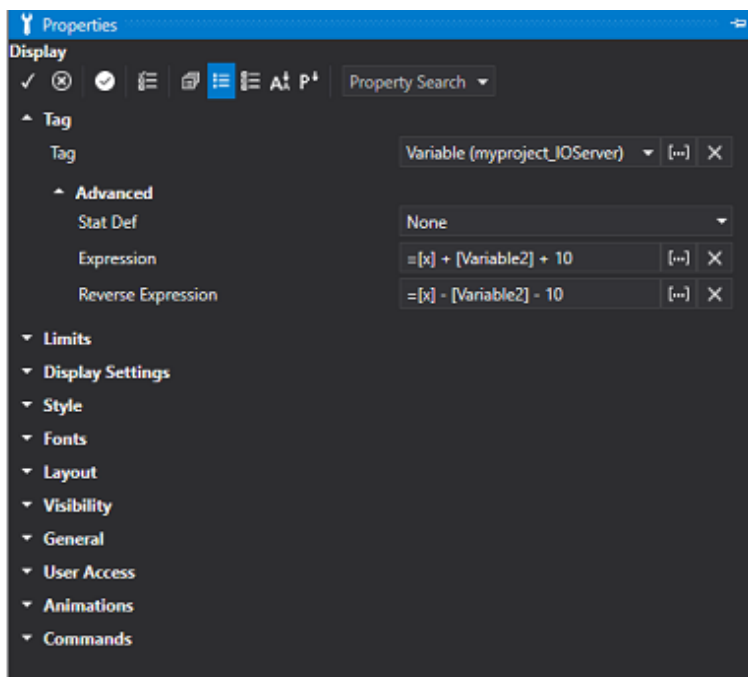
$$20 - 2 - 10 = 8$$



The inverse expression is valued only when the direct Expression has been inserted. Otherwise the Inverse Expression will be ignored.

To realize the above described example, please proceed as follows:

1. Open a screen and insert an object such as the Display object.
2. Assign the "TagName1" variable to the Display to use as the Display's Contextual Tag.
3. Insert the expression with the  $=[x] + [TagName2] + 10$  syntax in the Display's 'Expression' property.
4. Insert the expression with the  $=[x] - [TagName2] - 10$  syntax in the Display's 'Inverse Expression' property.
5. The Display will display the result of the expression at project runtime.



*Object Tooltip commands are used for inserting expressions or formulae for displaying data.*



Warning! A localization problem in converting double values currently exists. To remedy this problem for the time being you will need to set the English language in your local workstation in order to test expressions.

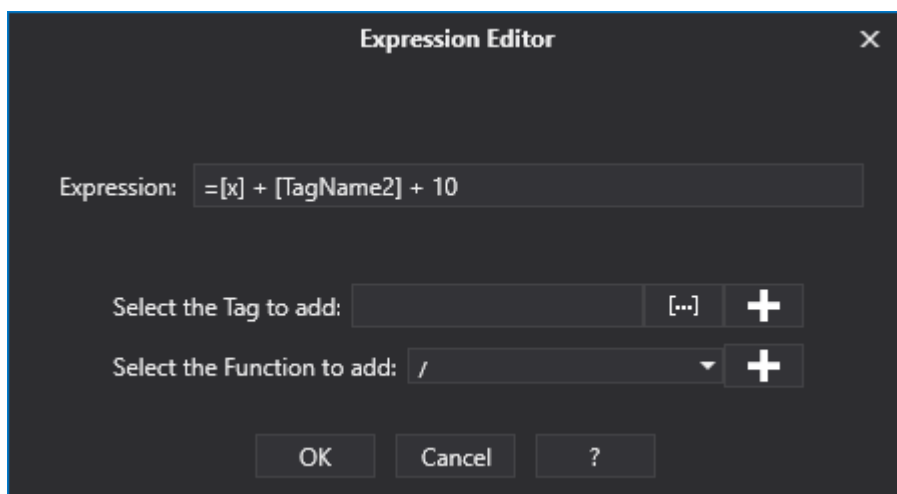


Attention! The expression manager does not recognize variable names as "Case Sensitive". Therefore the expression might not get valued correctly in cases where more than one variable have the same name but are distinguished by uppercase or lowercase letters (e.g. "TankLevel" and "tanklevel").

## Expression Syntax

The functions and syntaxes available for the expressions are Ms Excel 2013 type and permit mathematical and logic combinations with project tags.

This editor window will help you compose complex expressions.



Variable or formulae and functions can be added to the expression field by selecting them from the drop-down list indicated by the down pointed arrows on the right. After selecting the ones desired use the "Add" buttons to add them to Expression field above.



Warning! If a decimal value is inserted in the expression (eg 1.3 etc.) it will be necessary to use the dot (".") as decimal separator, independently from the Windows settings (eg: = [x] + [TagName2] + 1.3).

### List of functions that can be used in Expressions

FUNCTION	Syntax	Description
<b>ABS</b>	<b>ABS(number)</b>	Returns the absolute value of a number. The absolute value of a non-negative number is the number itself. The absolute value of a negative number is -1 times the number.
<b>ACCRINT</b>	<b>ACCRINT(issue, first_interest, settlement, rate, par, frequency, [basis], [calc_method])</b>	Returns the accrued interest for a security that pays periodic interest on a periodic basis.
<b>ACOS</b>	<b>ACOS(number)</b>	Returns the inverse cosine of a number. Inverse cosine is also referred to as arccosine. The arccosine is the angle whose cosine is the given number. The returned angle is given in radians in the range of 0 to pi.
<b>ACOSH</b>	<b>ACOSH(number)</b>	Returns the inverse hyperbolic cosine of a number. The number must be greater than or equal to 1. The inverse hyperbolic cosine is the value whose hyperbolic cosine is the given number.
<b>ACOT</b>	<b>ACOT(number)</b>	Retrieves the principal value of the inverse trigonometric cotangent of a number.
<b>ACOTH</b>	<b>ACOTH(number)</b>	Retrieves the inverse hyperbolic cotangent of a number.
<b>ACSCH</b>	<b>ACSCH(number)</b>	Returns the inverse hyperbolic cosecant
<b>AND</b>	<b>AND(logical1, logical2, ...)</b>	Returns <b>True</b> if all the arguments have a logical value of True and returns False if at least one argument is False.
<b>ARABIC</b>	<b>ARABIC( romannumeral )</b>	A Roman numeral is converted into an Arabic numeral.
<b>ASC</b>	<b>ASC(text)</b>	The ASC function used to converts full-width (double-byte) characters to half-width (single-byte) characters.

<b>ASECH</b>	<b>Y = ASECH(X)</b>	Inverse hyperbolic secant
<b>ASIN</b>	<b>ASIN(number)</b>	Returns the inverse sine of a number. Inverse sine is also referred to as arcsine. The arcsine is the angle whose sine is the given number. The returned angle is given in radians in the range from $-\pi/2$ to $+\pi/2$ .
<b>ASINH</b>	<b>ASINH(number)</b>	Returns the inverse hyperbolic sine of a number. The inverse hyperbolic sine is the value whose hyperbolic sine is the given number, so $\text{ASINH}(\text{SINH}(\text{number}))$ equals number.
<b>ATAN</b>	<b>ATAN(number)</b>	Returns the inverse tangent of a number. Inverse tangent is also known as arctangent. The arctangent is the angle whose tangent is a number. The returned angle is given in radians in the range from $-\pi/2$ to $+\pi/2$ .
<b>ATAN2</b>	<b>ATAN2(x_num,y_num)</b>	Returns the inverse tangent of the specified x and y co-ordinates. The arctangent is the angle from the x-axis to a line containing the origin (0, 0) and the point (x_num, y_num). The angle is given in radians between $-\pi$ and $\pi$ , excluding $-\pi$ .
<b>ATANH</b>	<b>ATANH(number)</b>	Returns the inverse hyperbolic tangent of a number. Number must be strictly between -1 and 1. The inverse hyperbolic tangent is the value whose hyperbolic tangent is number, so $\text{ATANH}(\text{TANH}(\text{number}))$ equals the given number.
<b>AVEDEV</b>	<b>AVEDEV(number1, number2, ...)</b>	Returns the average of the absolute mean deviations of data points. AVEDEV is a measure of the variability in a data set.
<b>AVERAGE</b>	<b>AVERAGE(number1, number2, ...)</b>	Returns the average (arithmetic mean) of the arguments.
<b>AVERAGEA</b>	<b>AVERAGEA(value1, value2,...)</b>	Calculates the average (arithmetic mean) of the values in the list of arguments. In addition to numbers and text logical values such as True and False are also included in the calculation.
<b>AVG</b>	<b>AVG(number1, number2,...)</b>	Returns the average (arithmetic mean) of the arguments.

<b>BINOMDIST</b>	<b>BINOMDIST(number_s, trials, probability_s, cumulative)</b>	Returns the individual term binomial distribution probability.
<b>BASE</b>	<b>BASE(number, radix, [min_length])</b>	The BASE function converts a number into a text representation with the given radix (base).
<b>BESSELI</b>	<b>BESSELI(X, N)</b>	Returns the modified Bessel function $I_n(X)$ , which is equivalent to the Bessel function evaluated for purely imaginary arguments.
<b>BESSELIJ</b>	<b>BESSELJ( X, N )</b>	Returns the Bessel function $J_n(X)$ .
<b>BESSELK</b>	<b>BESSELK( X, N )</b>	Returns the modified Bessel function $K_n(X)$ , which is equivalent to the Bessel function evaluated for purely imaginary arguments.
<b>BESSELY</b>	<b>BESSELY( X, N )</b>	Returns the Bessel function $Y_n(X)$ .
<b>BETA.DIST</b>	<b>BETA.DIST(x,alpha,beta,cumulative,[A],[B])</b>	Calculates the cumulative beta distribution function or the probability density function of the Beta distribution, for a given set of parameters.
<b>BIGMUL</b>	<b>BIGMUL(number1,number2)</b>	Returns the complete product of two numbers
<b>BIN2DEC</b>	<b>BIN2DEC(number)</b>	The BIN2DEC function converts a binary number into a decimal number.
<b>BIN2HEX</b>	<b>BIN2HEX(number places)</b>	The BIN2HEX function converts a binary number into a hexadecimal.
<b>BIN2OCT</b>	<b>BIN2OCT(num, places)</b>	The BIN2OCT function converts a binary number into an octal number.
<b>BINOM.DIST</b>	<b>BINOM.DIST (trial number,sp,value, cumulative)</b>	The Binom.Dist function returns the Binomial Distribution probability for a given number of successes from a specified number of trials.
<b>BINOM.INV</b>	<b>BINOM.INV(trial number,sp,value)</b>	The Binom.Inv function returns the smallest value for which the cumulative binomial distribution is greater than or equal to a criterion value.
<b>BINOMDIST</b>	<b>BINOMDIST(number_s,trials,probability_s,cumulative)</b>	Returns the individual term binomial distribution probability.

<b>BITAND</b>	<b>BITAND(expr1, expr2)</b>	Returns the output as bitwise AND of the inputs, the inputs, and its output are treated as vectors of bits.
<b>BITLSHIFT</b>	<b>BITLSHIFT(num1,num2)</b>	The BITLSHIFT function returns a number shifted left by specified number of bits.
<b>BITOR</b>	<b>BITOR(num1, num2)</b>	The BITOR function returns a bitwise 'OR' of two numbers.
<b>BITRSHIFT</b>	<b>BITRSHIFT(num1,num2)</b>	The BITRSHIFT function returns a number shifted right by the specified number of bits.
<b>BITXOR</b>	<b>BITXOR(num1, num2)</b>	The BITXOR function returns bitwise XOR of two numbers.
<b>CEILING</b>	<b>CEILING(number, significance)</b>	Returns number rounded up, away from zero, to the nearest multiple of significance. For example, if you want to avoid using pennies in your prices and your product is priced at \$4.82, use the formula =CEILING(4.82,0.05) to round prices up to the nearest nickel.
<b>CEILING.MATH</b>	<b>CEILING.MATH(number, [significance], [mode])</b>	Rounds a number up to the nearest integer or to the nearest multiple of significance.
<b>CHAR</b>	<b>Char(number)</b>	The <b>Char</b> function returns the character whose number code is defined in the parameter.
<b>CHIDIST</b>	<b>CHIDIST(x, degrees_freedom)</b>	Returns the one-tailed probability of the chi-squared distribution. The $\chi^2$ distribution is associated with a $\chi^2$ test. Use the $\chi^2$ test to compare observed and expected values.
<b>CHIINV</b>	<b>CHIINV(probability, degrees_freedom)</b>	Returns the inverse of the one-tailed probability of the chi-squared ( $\chi^2$ ) distribution. If probability = CHIDIST(x,...), then CHIINV(probability,...) = x. Use this function to compare observed results with expected ones in order to decide whether your original hypothesis is valid.
<b>CHISQ.DIST</b>	<b>CHISQ.DIST(x,degFreedom,cumulative)</b>	The Chisq.Dist function calculates the Probability Density Function or the Cumulative Distribution Function for the chi-square distribution.
<b>CHISQ.DIST.RT</b>	<b>CHISQ.DIST.RT(x,degFreedom)</b>	The Chisq.Dist.Rt function calculates the right-tailed probability of the chi-square distribution.

<b>CHISQ.INV</b>	<b>CHISQ.INV(probability,degFreedom)</b>	The Chisq.Inv function returns the inverse of the left-tailed probability of the chi-squared distribution.
<b>CHISQ.INV.RT</b>	<b>CHISQ.INV.RT(probability, degFreedom)</b>	The Chisq.Inv.Rt function calculates the inverse of the right-tailed probability of the chi-square distribution.
<b>CHISQ.TEST</b>	<b>CHISQ.TEST(actual_range,expected_range)</b>	Returns the chi-squared statistical test for independence
<b>CHITEST</b>	<b>CHITEST(actual_range, expected_range)</b>	Returns the test for independence. CHITEST returns the value from the chi-squared (c2) distribution for the statistic and the appropriate degrees of freedom.
<b>CLEAN</b>	<b>CLEAN(Text)</b>	The Clean function is used to remove the non-printable characters from the given text, represented by numbers 0 to 31 of the 7-bit ASCII code.
<b>CODE</b>	<b>CODE(name)</b>	The Code function converts the first character of a supplied text string into a numeric character set code.
<b>CHOOSE</b>	<b>Choose(index, valuearray)</b>	The <b>Choose</b> function returns the value from a range of values on a specific index.
<b>COLUMN</b>	<b>Column(range)</b>	The <b>Column</b> function returns the column index of the provided column in range.
<b>COMBIN</b>	<b>COMBIN(number, number_chosen)</b>	Returns the number of combinations for a given number of items. Use COMBIN to determine the total possible number of groups for a given number of items.
<b>COMBINA</b>	<b>COMBINA(number1, number2)</b>	For a given number of items, the Combina function returns the number of combinations.
<b>COMPLEX</b>	<b>COMPLEX(real_num, i_num, [suffix])</b>	The function COMPLEX converts user-supplied real and imaginary coefficients into a complex number
<b>CONCATENATE</b>	<b>CONCATENATE (text1, text2,...)</b>	Joins several text strings into one text string.
<b>CONFIDENCE</b>	<b>CONFIDENCE(alpha,standard_dev,size)</b>	Returns a value that you can use to construct a confidence interval about a population mean. The confidence interval is a range of values. In your sample, mean x is at the center

		of this range and the range is $x \pm \text{CONFIDENCE}$ . For example, if $x$ is the sample mean of delivery times for products ordered through the mail, $x \pm \text{CONFIDENCE}$ is a range of population means.
<b>CONFIDENCE.NORM</b>	<b>CONFIDENCE.NORM(alpha,stdev,size)</b>	The Confidence.Norm function uses Normal Distribution to calculate a confidence value that can be used to construct the confidence interval for a population mean, for a supplied probability, and sample size.
<b>CONFIDENCE.T</b>	<b>CONFIDENCE.T(alpha,standard_dev,size)</b>	Returns the confidence interval for a population mean
<b>CONVERT</b>	<b>CONVERT(number,from_unit,to_unit)</b>	The function CONVERT converts a number from one measurement system to another.
<b>CORREL</b>	<b>CORREL(array1, array2)</b>	Returns the correlation coefficient of the array1 and array2 cell ranges.
<b>COS</b>	<b>COS(number)</b>	Returns the cosine of the given angle.
<b>COSH</b>	<b>COSH(number)</b>	Returns the hyperbolic cosine of a number.
<b>COT</b>	<b>COT(number)</b>	The Cot function returns the cotangent of an angle specified in radians.
<b>COTH</b>	<b>COTH(number)</b>	The COTH function returns the hyperbolic cotangent of a hyperbolic angle.
<b>COUNT</b>	<b>COUNT(value1, value2,...)</b>	Counts the number of items in a list that contains numbers.
<b>COUNTA</b>	<b>COUNTA(value1, value2,...)</b>	Counts the number of cells that are not empty.
<b>COUNTBLANK</b>	<b>COUNTBLANK(range)</b>	Counts empty cells in a specified range of cells.
<b>COUNTIF</b>	<b>COUNTIF(range, criteria)</b>	Counts the number of cells within a range that meet the given criteria.
<b>COVAR</b>	<b>COVAR(array1, array2)</b>	Returns covariance, the average of the products of deviations for each data point pair.
<b>COVARIANCE.P</b>	<b>COVARIANCE.P(array1,array2)</b>	Returns population covariance, the average of the products deviation for each data point pair in two data sets.



<b>COVARIANCE.S</b>	<b>COVARIANCE.S(array1,array2)</b>	Returns the sample covariance, the average of the products deviation for each data point pair in two data sets.
<b>CRITBINOM</b>	<b>CRITBINOM(trials, probability_s, alpha)</b>	Returns the smallest value for which, the cumulative binomial distribution is greater than or equal to a criterion value.
<b>CSC</b>	<b>CSC(number)</b>	The CSC function returns the cosecant of an angle specified in radians.
<b>CSCH</b>	<b>CSCH(number)</b>	The CSCH function returns the hyperbolic cosecant of an angle specified in radians.
<b>CUMIPMT</b>	<b>CUMIPMT(rate, nper, pv, start_period, end_period, type)</b>	Calculates the cumulative interest paid between two specified periods.
<b>CUMPRINC</b>	<b>CUMPRINC(rate, nper, pv, start_period, end_period, type)</b>	Calculates the cumulative principal paid on a loan, between two specified periods.
<b>DATE</b>	<b>DATE(year, month, day)</b>	Returns the sequential serial number that represents a particular date.
<b>DATEVALUE</b>	<b>DATEVALUE(date_text)</b>	Returns the serial number of the date represented by the date_text.
<b>DAY</b>	<b>DAY(serial_number)</b>	Returns the day of a date, represented by a serial number. The day is given as an integer ranging from 1 to 31.
<b>DAYS</b>	<b>DAYS(endDate, startDate)</b>	The Days function retrieves the number of days between two dates.
<b>DAYS360</b>	<b>DAYS360(start_date, end_date, method)</b>	Returns the number of days between two dates based on a 360-day year (twelve 30-day months) which, is used in some accounting calculations.
<b>DB</b>	<b>DB(cost, salvage, life, period, month)</b>	Returns the depreciation of an asset for a specified period using the fixed-declining balance method.
<b>DCOUNT</b>	<b>DCOUNT(database, field, criteria)</b>	Returns the number of cells containing numbers in a field of a list or database that satisfy specified conditions.
<b>DCOUNTA</b>	<b>DCOUNTA(database, field, criteria)</b>	Returns the number of non-blank cells in a field of a list or database, that satisfy specified conditions.

<b>DDB</b>	<b>DDB(cost, salvage, life, period, factor)</b>	Returns the depreciation of an asset for a specified period using the double-declining balance method or some other method you specify.
<b>DEC2BIN</b>	<b>DEC2BIN(num,places)</b>	The DEC2BIN function converts a decimal number into a binary number.
<b>DEC2HEX</b>	<b>DEC2HEX(number, [places])</b>	The DEC2HEX function converts a decimal number to hexadecimal
<b>DEC2OCT</b>	<b>DEC2OCT(num, places)</b>	The DEC2OCT function converts a decimal number into an octal number.
<b>DECIMAL</b>	<b>DECIMAL(text, radix)</b>	A text representation of a number in a given base to be converted into a decimal number.
<b>DEGREES</b>	<b>DEGREES(angle)</b>	Converts radians into degrees.
<b>DELTA</b>	<b>DELTA(number1, [number2])</b>	The function DELTA tests whether two supplied numbers are equal.
<b>DEVSQ</b>	<b>DEVSQ(number1, number2,...)</b>	Returns the sum of squares of deviations of data points from their sample mean.
<b>DGET</b>	<b>DGET(database, field, criteria)</b>	Returns a single value from a field of a list or database, that satisfy specified conditions.
<b>DISC</b>	<b>DISC(settlement, maturity, pr, redemption, [basis])</b>	It will calculate the rate of discount for a bond. It helps to know the discount rate when we know the other details about the bond.
<b>DIVREM</b>	<b>DivRem(int dividend, int divisor, int result)</b>	This method is used to calculate the quotient of two 32-bit signed integers and returns the remainder in an output parameter.
<b>DMAX</b>	<b>DMAX(database, field, criteria)</b>	Returns the largest number in a field (column) of records in a list or database that matches conditions you that specify.
<b>DMIN</b>	<b>DMIN(database, field, criteria)</b>	Returns the smallest number in a field (column) of records in a list or database that matches conditions that you specify.
<b>DOLLAR</b>	<b>Dollar (number, decimal_places)</b>	The <b>Dollar</b> function converts a number to text, using a currency format. The format used is \$#,##0.00_);(\$#,##0.00).

<b>DOLLARDE</b>	<b>DOLLARDE(fractional-dollar, fraction)</b>	The DOLLARDE function is one of the financial functions. It is used to convert a dollar price represented as a fraction into a dollar price represented as a decimal number.
<b>DOLLARFR</b>	<b>DOLLARFR(decimal_dollar, fraction)</b>	DOLLARFR helps in converting a dollar value in decimal notation into a dollar value expressed in fractional notation.
<b>DSTDEV</b>	<b>DSTDEV(database, field, criteria)</b>	Estimates the standard deviation of a population based on a sample by using the numbers in a field (column) of records in a list or database that match conditions that you specify.
<b>DSTDEVP</b>	<b>DSTDEVP(database, field, criteria)</b>	Calculates the standard deviation (based on an entire population) of values in a field of a list or database, that satisfy specified conditions
<b>DSUM</b>	<b>DSUM(database, field, criteria)</b>	Adds the numbers in the field (column) of records in a list or database that match the conditions you specify
<b>DURATION</b>	<b>DURATION(settlement, maturity, coupon, yld, frequency, [basis])</b>	The DURATION function, one of the Financial functions, returns the Macauley duration for an assumed par value of \$100. Duration is defined as the weighted average of the present value of cash flows, and is used as a measure of a bond price's response to changes in yield.
<b>DVAR</b>	<b>DVAR(database, field, criteria)</b>	Calculates the variance (based on a sample of a population) of values in a field of a list or database, that satisfy specified conditions.
<b>DVARP</b>	<b>DVARP(database, field, criteria)</b>	Calculates the variance (based on an entire population) of values in a field of a list or database, that satisfy specified conditions.
<b>EDATE</b>	<b>EDATE( startDate, Months )</b>	The EDate function returns a date that is a specified number of months before or after a supplied start date.
<b>ENCODEURL</b>	<b>ENCODEURL(name)</b>	The EncodeURL function retrieves an URL-encoded string.
<b>EOMONTH</b>	<b>EOMONTH(startDate, Months)</b>	The EOMONTH function returns the last day of the month that is a specified number of months

		before or after an initially supplied start date.
<b>ERF</b>	<b>ERF(lower_limit,[upper_limit])</b>	The function ERF returns the error function integrated between two supplied limits
<b>ERF.PRECISE</b>	<b>ERF.PRECISE(x)</b>	The function ERF.PRECISE returns the error function integrated between 0 and a supplied limit
<b>ERFC.PRECISE</b>	<b>ERFC.PRECISE(x)</b>	The function ERFC.PRECISE returns the complementary error function integrated between a supplied lower limit and infinity.
<b>ERROR.TYPE</b>	<b>= ERROR.TYPE(value)</b>	Error.Type function returns an integer for the given error value that denotes the type of given error.
<b>EVEN</b>	<b>EVEN(number)</b>	Returns the number rounded up to the nearest even integer.
<b>EXACT</b>	<b>Exact(value1, value2)</b>	The <b>Exact</b> function compares two values ignoring the styles and returns the boolean value as true or false.
<b>EXP</b>	<b>EXP(number)</b>	Returns e raised to the power of the given number.
<b>EXPON.DIST</b>	<b>EXPON.DIST(x,y,cumulative)</b>	The Expon.Dist function calculates the value of the probability density function or the cumulative distribution function for the <b>Exponential Distribution</b> .
<b>EXPONDIST</b>	<b>EXPONDIST(x, lambda, cumulative)</b>	Returns the exponential distribution.
<b>F.DIST</b>	<b>F.DIST(x,degFreedom1,degFreedom2,cumulative)</b>	The F.Dist function calculates the Probability Density Function or the Cumulative Distribution Function for the F Distribution.
<b>F.DIST.RT</b>	<b>F.DIST.RT(x, degFreedom1, degFreedom2)</b>	The F.Dist.Rt function calculates the F Probability Distribution that measures the degree of diversity between two data sets.
<b>F.INV.RT</b>	<b>F.INV.RT(probability,degFreedom1,degFreedom2)</b>	The F.INV.RT function calculates the inverse of the Cumulative F Distribution for a supplied probability.
<b>FACT</b>	<b>FACT(number)</b>	Returns the factorial of a number. The factorial of a number is the product of all positive integers <= the given number.

<b>FACTDOUBLE</b>	<b>FACTDOUBLE (number)</b>	FactDouble function returns the double factorial of a given value. The given value is an integer value.
<b>FALSE</b>	<b>False(stringvalue)</b>	The <b>False</b> function returns the logical value for the false.
<b>FDIST</b>	<b>FDIST(x, degrees_freedom1, degrees_freedom2)</b>	Returns the F probability distribution.
<b>FILTERXML</b>	<b>FILTERXML(xml, xpath)</b>	The Filterxml function returns specific data from XML content by using the specified xpath.
<b>FIND</b>	<b>Find(lookfor, lookin, start)</b>	The <b>Find</b> finction finds a portion of a string from a particular text and returns the location of the string.
<b>FINDB</b>	<b>FINDB(find_text, within_text, [start_num])</b>	FINDB locate one text string within a second text string, and return the number of the starting position of the first text string from the first character of the second text string.
<b>FINV</b>	<b>FINV(probability,deg_freedom1,deg_freedom2)</b>	The <b>Finv</b> function returns the inverse of the F probability distribution. If $p = \text{FDIST}(x, \dots)$ , then $\text{FINV}(p, \dots) = x$ . Using F distribution, you can compare the degree of variability for two data sets.
<b>FISHER</b>	<b>FISHER(x)</b>	Returns the Fisher transformation at x. This transformation produces a function that is normally distributed rather than skewed.
<b>FISHERINV</b>	<b>FISHERINV(y)</b>	Returns the inverse of the Fisher transformation. If $y = \text{FISHER}(x)$ , then $\text{FISHERINV}(y) = x$ .
<b>FIXED</b>	<b>Fixed ( number, decimal_places, no_commas )</b>	The <b>Fixed</b> function rounds off to a specified number of decimal places and returns the value in text format.
<b>FLOOR</b>	<b>FLOOR(number, significance)</b>	Rounds off the given number down, toward zero, to the nearest multiple of significance.
<b>FORECAST</b>	<b>FORECAST(x, known_ys, known_xs)</b>	Calculates a future value by using existing values using a linear regression. The predicted value is a y-value for a given x-value.
<b>FORMULATEXT</b>	<b>FORMULATEXT (reference)</b>	FormulaText function returns the formula as a string.

<b>FV</b>	<b>FV( interest_rate, number_payments, payment, PV, Type )</b>	The <b>FV</b> function returns the future value of an investment, based on an interest rate and a constant payment schedule.
<b>FVSCHEDULE</b>	<b>FVSCHEDULE(principal, schedule)</b>	Calculates the future value of an initial principal, after applying a series of compound interest rates.
<b>GAMMA.DIST</b>	<b>GAMMA.DIST(x,alpha,beta,cumulative)</b>	Returns the gamma distribution
<b>GAMMA.INV</b>	<b>GAMMA.INV(x,y,z,cumulative)</b>	The Gamma.Inv function returns the inverse of the Gamma Distribution.
<b>GAMMADIST</b>	<b>GAMMADIST(x, alpha,beta, cumulative)</b>	Returns the gamma distribution.
<b>GAMMAINV</b>	<b>Gammainv(p, alpha, beta)</b>	The <b>Gammainv</b> function returns the inverse function for the GAMMADIST function.
<b>GAMMAINV</b>	<b>GAMMAINV(probability, alpha, beta)</b>	Returns the inverse of the gamma cumulative distribution. If p = GAMMADIST(x,...), then GAMMAINV(p,...) = x.
<b>GAMMALN</b>	<b>GAMMALN(x)</b>	Calculates the natural logarithm of the gamma function for a supplied value.
<b>GAMMALN.PRECISE</b>	<b>GAMMALN.PRECISE(x)</b>	The Gammaln.Precise function returns the natural logarithm of the Gamma Distribution.
<b>GCD</b>	<b>GCD(number1, [number2], ...)</b>	The GCD function returns the greatest common divisor of two or more integers. The greatest common divisor is the largest integer that that goes into all given numbers without a remainder.
<b>GEOMEAN</b>	<b>GEOMEAN(number1, number2,...)</b>	Returns the geometric mean of an array or range of positive data.
<b>GESTEP</b>	<b>GESTEP(number, [step])</b>	The function GESTEP tests whether a number is greater than a supplied threshold value.
<b>GROWTH</b>	<b>=GROWTH(known_y's, [known_x's], [new_x's],</b>	This feature enables you to calculate predicted exponential growth using existing data. This calculates and returns an array of values used for the regression analysis. Growth enables you to perform a regression analysis.
<b>HARMEAN</b>	<b>HARMEAN(number1, number2,...)</b>	Returns the harmonic mean of a data set. The harmonic mean is the reciprocal of the arithmetic mean of reciprocals.

<b>HEX2BIN</b>	<b>HEX2BIN(num, places)</b>	The HEX2BIN function converts a hexadecimal number into a binary number.
<b>HEX2DEC</b>	<b>HEX2DEC(number)</b>	The HEX2DEC function Converts a hexadecimal number to a decimal.
<b>HEX2OCT</b>	<b>HEX2OCT(num, places)</b>	The HEX2OCT function converts a hexadecimal number into an octal number.
<b>HLOOKUP</b>	<b>HLOOKUP(lookup_value, table_array, row_index_num, range_lookup)</b>	Searches for a value in the top row of the array of values and then returns a value in the same column from a row you specify in the array. Use HLOOKUP when your comparison values are located in a row across the top of a table of data and you want to look down a specified number of rows. Use VLOOKUP when your comparison values are located in a column to the left of the data you want to find.
<b>HOURL</b>	<b>HOURL(serial_number)</b>	Returns the hour of a time value. The hour is given as an integer, ranging from 0 (12:00 A.M.) to 23 (11:00 P.M.).
<b>HYPERLINK</b>	<b>HYPERLINK(linkLocation, friendlyName)</b>	The Hyperlink function creates a hyperlink to a document in a supplied location.
<b>HYPGEOM.DIST</b>	<b>HYPGEOM.DIST(sample_s,number_sample,population_s,number_pop,cumulative)</b>	Returns the hypergeometric distribution.
<b>HYPGEOMDIST</b>	<b>Hypgeomdist(sample, numberofsample, population, numberofpopulation)</b>	The <b>Hypgeomdist</b> function returns the hypergeometric distribution.
<b>HYPEGEOMDIST</b>	<b>HYPGEOM.DIST(sample_s,number_sample,population_s,number_pop,cumulative)</b>	Returns the hypergeometric distribution. HYPGEOMDIST returns the probability of a given number of sample successes, given the sample size, population successes and population size.
<b>IF</b>	<b>IF(logical_test, value_if_true, value_if_false)</b>	Returns one value if a condition you specify evaluates to True and another value if it evaluates to False. Use IF to conduct conditional tests on values and formulas.
<b>IFERROR</b>	<b>IFERROR (value, value_error)</b>	IfError function tests when an initial given value (or expression) returns an error, and then, this function returns a second given argument. Otherwise, the function returns the initial tested value.

<b>IFNA</b>	<b>IFNA (Formula_value, value_if_na)</b>	IFNA function returns the value specified when the formula returns the #N/A error value; otherwise, it returns the result of the given formula.
<b>IMABS</b>	<b>IMABS(inumber)</b>	The IMABS function returns the absolute value (the modulus) of a complex number.
<b>IMAGINARY</b>	<b>IMAGINARY(inumber)</b>	The IMAGINARY function returns the imaginary coefficient of a complex number.
<b>IMARGUMENT</b>		
<b>IMCONJUGATE</b>	<b>IMCONJUGATE(inumber)</b>	The function IMCONJUGATE returns the complex conjugate of a complex number.
<b>IMCOS</b>	<b>IMCOS(inumber)</b>	The function IMCOS returns the cosine of a complex number
<b>IMCOSH</b>	<b>IMCOSH(inumber)</b>	The function IMCOSH returns the hyperbolic cosine of a complex number.
<b>IMCOT</b>	<b>IMCOT(inumber)_</b>	The function IMCOT returns the cotangent of a complex number.
<b>IMCOTH</b>	<b>IMCOTH(number)</b>	The IMCOTH function returns the hyperbolic cotangent of the given complex number. For example, a given complex number "x+yi" returns "coth(x+yi)."
<b>IMCSC</b>	<b>IMCSC(inumber)</b>	The function IMCSC returns the cosecant of a complex number
<b>IMCSCH</b>	<b>IMCSCH(inumber)</b>	The function IMCSCH returns the hyperbolic cosecant of a complex number.
<b>IMDIV</b>	<b>IMDIV(inumber1, inumber2)</b>	The function IMDIV returns the quotient of two supplied complex numbers.
<b>IMEXP</b>	<b>IMEXP(inumber)</b>	The function IMEXP returns the exponential of a complex number.
<b>IMLN</b>	<b>IMLN(inumber)</b>	The function IMLN returns the natural logarithm of a complex number.
<b>IMLOG10</b>	<b>IMLOG10(inumber)</b>	The function IMLOG10 returns the base-10 logarithm of a complex number.



<b>IMLOG2</b>	<b>IMLOG2(inumber)</b>	The function IMLOG2 returns the base-2 logarithm of a complex number.
<b>IMPOWER</b>	<b>IMPOWER(inumber, number)</b>	The function IMPOWER calculates a complex number raised to a supplied power.
<b>IMPRODUCT</b>	<b>IMPRODUCT(inumber1, [inumber2], ...)</b>	The function IMPRODUCT returns the product of up to 255 supplied complex numbers.
<b>IMREAL</b>	<b>IMREAL(inumber)</b>	The function IMREAL function returns the real coefficient of a complex number.
<b>IMSEC</b>	<b>IMSEC(inumber)</b>	The function IMSEC returns the secant of a complex number.
<b>IMSECH</b>	<b>IMSECH(inumber)</b>	The function IMSECH returns the hyperbolic secant of a complex number.
<b>IMSIN</b>	<b>IMSIN(inumber)</b>	The function IMSIN returns the sine of a complex number.
<b>IMSINH</b>	<b>IMSINH(inumber)</b>	The function IMSINH returns the hyperbolic sine of a complex number.
<b>IMSQRT</b>	<b>IMSQRT(inumber)</b>	The function IMSQRT returns the square root of a complex number.
<b>IMSUB</b>	<b>IMSUB(inumber1, inumber2)</b>	The function IMSUB subtracts two complex numbers
<b>IMSUM</b>	<b>IMSUM(inumber1, [inumber2], ...)</b>	The function IMSUM calculates the sum of two complex numbers
<b>IMTAN</b>	<b>IMTAN(inumber)</b>	The function IMTAN returns the tangent of a complex number.
<b>IMTANH</b>	<b>IMTANH(number)</b>	The IMTANH function returns the hyperbolic tangent of the given complex number. For example, a given complex number "x+yi" returns "tanh(x+yi)."
<b>INDEX</b>	<b>Index(range,row,col)</b>	The <b>Index</b> function returns the exact value from the provided row index and column index from a specific range.
<b>INDIRECT</b>	<b>Indirect(content)</b>	The <b>Indirect</b> function returns the reference as a string instead of providing the content or range within it.
<b>INFO</b>	<b>INFO(infoType)</b>	The Info function returns a text string containing information

		about the current operating environment.
<b>INT</b>	<b>INT(number)</b>	Rounds a number down to the nearest integer.
<b>INTERCEPT</b>	<b>INTERCEPT(known_y's, known_x's)</b>	Calculates the point at which, the least squares fit line will intersect the y-axis.
<b>IPMT</b>	<b>IPMT(rate, per, nper, pv, fv, type)</b>	Returns the interest payment for a given period for an investment based on periodic, constant payments and a constant interest rate.
<b>IRR</b>	<b>IRR(values, guess)</b>	Returns the internal rate of return for a series of cash flows represented by the numbers in values. The cash flows must occur at regular intervals such as monthly or annually.
<b>ISBLANK</b>	<b>IsBlank( value )</b>	The <b>IsBlank</b> function checks for blank or null values.
<b>ISERR</b>	<b>IsErr( value )</b>	value is the value that you want to test. If the value is an error value (except #N/A), this function will return TRUE/FALSE to indicate whether a value is an error.
<b>ISERROR</b>	<b>ISERROR(value)</b>	Returns True if the value is a string that starts with a #.
<b>ISEVEN</b>	<b>ISEVEN (value)</b>	The ISEVEN function returns TRUE if given number is an even number and returns FALSE if the given number is an odd number.
<b>ISFORMULA</b>	<b>ISFORMULA (reference)</b>	The IsFormula function returns true or false when there is a reference to a cell that contains a formula.
<b>ISLOGICAL</b>	<b>IsLogical( value )</b>	The <b>IsLogical</b> function checks whether a value is a logical value and returns a TRUE or FALSE.
<b>ISNA</b>	<b>IsNA(value)</b>	The <b>IsNA</b> function returns a boolean value after determining that the provided value is a #NA error value.
<b>ISNONTEXT</b>	<b>IsNonText(text)</b>	The <b>IsNonText</b> function returns the boolean value after determining that the provided value is not a string.
<b>ISNUMBER</b>	<b>ISNUMBER(value)</b>	Returns True if the value parses as a numeric value.

<b>ISODD</b>	<b>ISFORMULA (reference)</b>	IsOdd function returns true when the given number is an odd number and returns false when the given number is even.
<b>ISOWEEKNUM</b>	<b>ISOWEEKNUM( DateTime)</b>	For a given date, the ISOWeekNum function returns the ISO week number of that year.
<b>ISPMT</b>	<b>ISPMT(rate, per, nper, pv)</b>	Calculates the interest paid during a specific period of an investment.
<b>ISREF</b>	<b>ISREF(given_value)</b>	The ISREF function returns the logical value TRUE if the given value is a reference value; otherwise, the function returns FALSE.
<b>ISTEXT</b>	<b>IsText(text)</b>	The <b>IsText</b> function returns a boolean value after determining that the provided value is a string.
<b>JIS</b>	<b>JIS(text)</b>	The JIS function converts single byte characters into double byte characters in a text string.
<b>KURT</b>	<b>KURT(number1, number2, ...)</b>	Returns the kurtosis of a data set. Kurtosis characterizes the relative peakedness or flatness of a distribution compared with the normal distribution. Positive kurtosis indicates a relatively peaked distribution. Negative kurtosis indicates a relatively flat distribution.
<b>LARGE</b>	<b>LARGE(array, k)</b>	Returns the k-th largest value in a data set.
<b>LCM</b>	<b>LCM(number1, [number2], ...)</b>	The LCM function returns the least common multiple of integers. The least common multiple is the smallest positive integer that is a multiple of all given numbers.
<b>LEFT</b>	<b>LEFT(text, num_chars)</b>	LEFT returns the first character or characters in a text string, based on the number of characters you specify.
<b>LEFTB</b>	<b>LEFTB(text, bytes)</b>	The LEFTB function calculates the first character or characters in a text string, based on the number of bytes you specify.
<b>LN</b>	<b>LN(number)</b>	Returns the natural logarithm of a number. Natural logarithms are based on the constant e (2.718281828459...).

<b>LEN</b>	<b>Len(text)</b>	LEN returns the length of a text string, including spaces.
<b>LENB</b>	<b>LENB(name)</b>	The LENB function returns the number of bytes used to represent the characters in a text string.
<b>LOG</b>	<b>LOG(number, base)</b>	Returns the logarithm of a number to the base that you specify.
<b>LOG10</b>	<b>LOG10(number)</b>	Returns the base-10 logarithm of a number.
<b>LOGEST</b>	<b>=LOGEST(known_y's, [known_x's], [const], [stats])</b>	This feature enables you to calculate predicted exponential growth using existing data. This calculates and returns an array of values used for the regression analysis. Logest calculates and returns an array of values that is used in regression analysis.
<b>LOGINV</b>	<b>LOGINV(probability, mean, standard_dev)</b>	Returns the inverse of the lognormal cumulative distribution function of x, where $\ln(x)$ is normally distributed with parameters mean and standard_dev. If $p = \text{LOGNORMDIST}(x, \dots)$ , then $\text{LOGINV}(p, \dots) = x$ .
<b>LOGNORM.DIST</b>	<b>LOGNORM.DIST(x, mean, stdev, cumulative)</b>	The Lognorm.Dist function calculates the Log-Normal Probability Density Function or the Cumulative Log-Normal Distribution Function for a supplied value of x.
<b>LOGNORM.INV</b>	<b>LOGNORM.INV(probability, mean, stdev)</b>	The Lognorm.Inv function calculates the inverse of the Cumulative Log-Normal Distribution Function of x for a supplied probability.
<b>LOGNORMDIST</b>	<b>LOGNORMDIST(x, mean, standard_dev)</b>	Returns the cumulative lognormal distribution of x, where $\ln(x)$ is normally distributed with parameters mean and standard_dev.
<b>LOWER</b>	<b>Lower( text )</b>	The <b>Lower</b> function converts all characters in the specified text string to lowercase. Characters in the string that are not letters, are not changed.
<b>MATCH</b>	<b>Match( value, array, match_type )</b>	The <b>Match</b> function searches for a specified value in an array and returns the relative position of that item.

<b>MAX</b>	<b>MAX(number1, number2, ...)</b>	Returns the largest value in a set of values.
<b>MAXA</b>	<b>MAXA(value1, value2, ...)</b>	Returns the largest value in a list of arguments. Text and logical values such as True and False are compared as well as numbers.
<b>MEDIAN</b>	<b>MEDIAN(number1, number2, ...)</b>	Returns the median of the given numbers. The median is the number in the middle of a set of numbers; that is, half the numbers have values that are greater than the median and half have values that are less.
<b>MID</b>	<b>MID(text, start_position, num_chars)</b>	MID returns a text segment of a character string. The parameters specify the starting position and the number of characters.
<b>MIDB</b>	<b>MIDB(text, startNum, numBytes)</b>	The MIDB function returns a specific number of characters from a text string, starting at the position you specify, based on the number of bytes you specify.
<b>MIN</b>	<b>MIN(number1, number2, ...)</b>	Returns the smallest number in a set of values.
<b>MINA</b>	<b>MINA(value1, value2, ...)</b>	Returns the smallest value in the list of arguments. Text and logical values such as True and False are compared as well as numbers.
<b>MINUTE</b>	<b>MINUTE(serial_number)</b>	Returns the minutes of a time value. The minute is given as an integer, ranging from 0 to 59.
<b>MIRR</b>	<b>MIRR(values, finance_rate, reinvest_rate)</b>	Returns the modified internal rate of return for a series of periodic cash flows.
<b>MOD</b>	<b>MOD(number, divisor)</b>	Returns the remainder after the number is divided by a divisor. The result has the same sign as the divisor.
<b>MODE</b>	<b>MODE(number1, number2, ...)</b>	Returns the most frequently occurring or repetitive, value in an array or range of data.
<b>MONTH</b>	<b>MONTH(serial_number)</b>	Returns the month of a date represented by a serial number. The month is given as an integer, ranging from 1 (January) to 12 (December).
<b>MROUND</b>	<b>MROUND(number, multiple)</b>	The MROUND function rounds a number up or down, depending on the nearest multiple of given number.

<b>MULTINOMIAL</b>	<b>MULTINOMIAL(number1, [number2], ...)</b>	The MULTINOMIAL function calculates the ratio of the factorial of a sum of values to the product of factorials of those values.
<b>N</b>	<b>N (value)</b>	The N function converts the given value into a numeric value.
<b>NA</b>	<b>NA()</b>	The NA function returns the #N/A error. This error message is produced when a formula is unable to find a value that it needs. This error message denotes 'value not available'.
<b>NEGBINOM.DIST</b>	<b>NEGBINOM.DIST(F_number,S_number,S_probability,cumulative)</b>	The Negbinom.Dist function calculates the probability mass function or the cumulative distribution function for the Negative Binomial Distribution.
<b>NETWORKDAYS.INTL</b>	<b>NETWORKDAYS.INTL(startDate, endDate)</b>	NETWORKDAYS.INTL calculates the number of whole work days between two supplied dates.
<b>NEGBINOMDIST</b>	<b>NEGBINOMDIST(number_f, number_s, probability_s)</b>	Returns the negative binomial distribution. NEGBINOMDIST returns the probability that there will be number_f failures before the number_s-th success, when the constant probability of a success is probability_s.
<b>NORM.DIST</b>	<b>NORM.DIST(x,mean,stdev,cumulative)</b>	The NORM.DIST function calculates the normal distribution for a supplied value of x, and a supplied distribution mean & standard deviation.
<b>NORM.INV</b>	<b>NORM.INV(probability,mean,standard_dev)</b>	Returns the inverse of the normal cumulative distribution
<b>NORM.S.DIST</b>	<b>NORM.S.DIST(val, cumulative)</b>	The Norm.S.Dist function returns the standard normal distribution.
<b>NORM.S.INV</b>	<b>NORM.S.INV(probability)</b>	The Norm.S.Inv function returns the inverse of the standard normal cumulative distribution.
<b>NORMDIST</b>	<b>NORMDIST(x, mean, standard_dev, cumulative)</b>	Returns the normal distribution for the specified mean and standard deviation.
<b>NORMINV</b>	<b>NORMINV(probability, mean, standard_dev)</b>	Returns the inverse of the normal cumulative distribution for the specified mean and standard deviation.
<b>NORMSDIST</b>	<b>NormsDist(value)</b>	The <b>NormsDist</b> function returns the probability that the observed value of a standard normal random variable will be less than or equal to the parameter.

<b>NORMSINV</b>	<b>NormsInv(value)</b>	The <b>NormsInv</b> function returns the standard normal random variable that has <i>Mean</i> 0 and <i>Standard Deviation</i> 1
<b>NOT</b>	<b>NOT(logical)</b>	Reverses the value of its argument.
<b>NOW</b>	<b>NOW( )</b>	Returns the serial number of the current date and time.
<b>NPER</b>	<b>NPER(rate, pmt, pv, fv, type)</b>	Returns the number of periods for an investment based on periodic, constant payments and a constant interest rate.
<b>NPV</b>	<b>NPV(rate, value1, value2, ...)</b>	Calculates the net present value of an investment by using a discount rate and a series of future payments (negative values) and income (positive values).
<b>NUMBERVALUE</b>	<b>NUMBERVALUE(text)</b>	The NumberValue function converts text to a number in a locale-independent way.
<b>OCT2BIN</b>	<b>OCT2BIN(number, [places])</b>	The OCT2BIN function Converts octal number to binary.
<b>OCT2DEC</b>	<b>OCT2DEC(number)</b>	The OCT2DEC function Converts octal number to a decimal.
<b>OCT2HEX</b>	<b>OCT2HEX(number, [places])</b>	The OCT2HEX function Converts octal number to hexadecimal.
<b>ODD</b>	<b>ODD(number)</b>	Returns the number rounded up to the nearest odd integer.
<b>OFFSET</b>	<b>Offset( range, rows, columns, height, width )</b>	The <b>Offset</b> function returns a reference to a range that is offset a number of rows and columns from any given range or cell.
<b>OR</b>	<b>OR(logical1, logical2, ...)</b>	Returns True if any argument is True; returns False if all arguments are <b>False</b> .
<b>PEARSON</b>	<b>PEARSON(array1, array2)</b>	Returns the Pearson product moment correlation coefficient, <i>r</i> , a dimensionless index that ranges from -1.0 to 1.0 inclusive and reflects the extent of a linear relationship between two data sets.
<b>PERCENTILE</b>	<b>PERCENTILE(array, k)</b>	Returns the k-th percentile of values in a range.
<b>PERCENTILE.EXC</b>	<b>PERCENTILE.EXC(array, k)</b>	The Percentile.Exc function returns the k-th percentile of

		values in a range, where k is in the range of 0 to 1 exclusively.
<b>PERCENTILE.INC</b>	<b>PERCENTILE.INC (array,k)</b>	The Percentile.Inc function returns the k-th percentile of values in a range, where k is in the range 0 to 1.
<b>PERCENTRANK</b>	<b>PERCENTRANK(array, x, significance)</b>	Returns the rank of a value in a data set as a percentage of the data set.
<b>PERCENTRANK.EXC</b>	<b>PERCENTRANK.EXC(array,x,[significance])</b>	Returns the rank of value in dataset as a percentage of the data set as percentage (0....1, exclusive) of the dataset
<b>PERCENTRANK.INC</b>	<b>PERCENTRANK.INC(array,x,[significance])</b>	Returns the rank of value in dataset as a percentage of the data set as percentage (0....1, inclusive) of the dataset
<b>PERMUT</b>	<b>Permut(n, k)</b>	The <b>Permut</b> function returns the number of permutations of n items taken at k time.
<b>PERMUTATIONA</b>	<b>PERMUTATIONA(number, number-chosen)</b>	The PermutationA function returns the number of permutations for a given number of objects (with repetitions) that can be selected from the total number of objects.
<b>PI</b>	<b>PI( )</b>	Returns the number 3.14159265358979, the mathematical constant pi, accurate to 15 digits.
<b>PMT</b>	<b>PMT(rate, nper, pv, fv, type)</b>	Calculates the payment for a loan based on constant payments and a constant interest rate.
<b>POISSON</b>	<b>POISSON(x, mean, cumulative)</b>	Returns the Poisson distribution.
<b>POISSON.DIST</b>	<b>POISSON.DIST(x,mean,cumulative)</b>	The Poisson.Dist function calculates the Poisson Probability Mass Function or the Cumulative Poisson Probability Function for a supplied set of parameters.
<b>POW</b>	<b>POW(number, power)</b>	The <b>Pow</b> function returns the result of a number raised to a power.
<b>POWER</b>	<b>POWER(number, power)</b>	Returns the result of a number raised to a power.
<b>PPMT</b>	<b>PPMT(rate, per, nper, pv, fv, type)</b>	Returns the payment on the principal for a given period, for an investment based on



		periodic, constant payments and a constant interest rate.
<b>PROB</b>	<b>PROB(x_range, prob_range, lower_limit, upper_limit)</b>	Returns the probability whose values are in a range that is between two limits. If upper_limit is not supplied, returns the probability that values in x_range are equal to lower_limit.
<b>PRODUCT</b>	<b>PRODUCT(number1, number2, ...)</b>	Multiplies all the numbers given as arguments and returns the product.
<b>PROPER</b>	<b>PROPER(text)</b>	The Proper function changes the first letter of text into a capital letter and following letters changes to lowercases.
<b>PV</b>	<b>PV(rate, nper, pmt, fv, type)</b>	Returns the present value of an investment. The present value is the total amount that a series of future payments is worth now.
<b>QUARTILE</b>	<b>QUARTILE(array, quart)</b>	Returns the quartile of a data set.
<b>QUOTIENT</b>	<b>QUOTIENT(numerator, denominator)</b>	The QUOTIENT function calculates the integer portion of a division. This function is used to discard the remainder of a division.
<b>RADIANS</b>	<b>RADIANS(angle)</b>	Converts degrees to radians.
<b>RAND</b>	<b>RAND( )</b>	Returns an evenly distributed random number greater than or equal to 0 and less than 1.
<b>RANDBETWEEN</b>	<b>RANDBETWEEN(bottom, top)</b>	The RANDBETWEEN function returns a random integer number between given numbers. A new random integer number is returned each time when worksheet is recalculated.
<b>RANK</b>	<b>RANK(number, ref, order)</b>	Returns the rank of a number in a list of numbers. The rank of a number is its size relative to other values in a list. (If you were to sort the list, the rank of the number would be its position.)
<b>RATE</b>	<b>RATE(nper, pmt, pv, fv, type, guess)</b>	Returns the interest rate per period of an annuity. RATE is calculated by iteration and may not converge to a unique solution.
<b>RECEIVED</b>	<b>RECEIVED(settlement, maturity, investment, discount, [basis])</b>	Calculates the amount received at maturity for a fully invested Security

<b>REPLACE</b>	<b>REPLACE(oldText, startNum, numChars, newText)</b>	The Replace function replaces a certain part of text with a different part of text based on the number of characters given.
<b>REPLACEB</b>	<b>REPLACEB(old_text, start_num, num_bytes, new_text)</b>	REPLACEB replaces part of a text string, based on the number of bytes you specify, with a different text string.
<b>REPT</b>	<b>REPT(string, number)</b>	The Rept function returns a supplied text string, repeated a specified number of times.
<b>RIGHT</b>	<b>RIGHT(text, num_chars)</b>	RIGHT returns the last character or characters in a text string, based on the number of characters you specify.
<b>RIGHTB</b>	<b>RIGHTB(string, num_bytes)</b>	The RIGHTB function returns the last character or characters in a text string, based on the number of bytes.
<b>ROMAN</b>	<b>ROMAN(number, [form])</b>	The ROMAN function converts an Arabic numeral to roman numeral as text.
<b>ROUND</b>	<b>ROUND(number, num_digits)</b>	Rounds a number to a specified number of digits.
<b>ROUNDDOWN</b>	<b>ROUNDDOWN(number, num_digits)</b>	Rounds a number down towards zero.
<b>ROUNDUP</b>	<b>ROUNDUP(number, num_digits)</b>	Rounds a number up away from 0 (zero).
<b>RRI</b>	<b>RRI(nper, pv, fv)</b>	Calculates an equivalent interest rate for the growth of an investment
<b>RSQ</b>	<b>RSQ(known_y's, known_x's)</b>	Returns the square of the Pearson product moment correlation coefficient through data points in known_y's and known_x's.
<b>SEARCH</b>	<b>SEARCH(findText,withinText, startNum)_</b>	The Search function finds one text string (find_text) within another text string (within_text), and returns the number of the starting position of find_text.
<b>SEARCHB</b>	<b>SEARCHB(findText,withinText, startNum)_</b>	The SEARCHB function returns the position of a specified character or sub-string within a given text string.
<b>SEC</b>	<b>SEC(number)</b>	The Sec function returns the secant of an angle.
<b>SECH</b>	<b>SECH(number)</b>	The SECH function returns the hyperbolic secant of an angle.

<b>SECOND</b>	<b>SECOND(serial_number)</b>	Returns the seconds of a time value. The second is given as an integer in the range 0 (zero) to 59.
<b>SERIESSUM</b>	<b>SERIESSUM(x, n, m, coefficients)</b>	Returns the sum of a power series based on the formula: $\text{SERIESSUM}(x, n, m, a) = a_1 * x^n + a_2 * x^{(n+m)} + a_i * x^{(n+(i-1)*m)}$
<b>SIGN</b>	<b>SIGN(number)</b>	Determines the sign of a number. Returns 1 if the number is positive, zero (0) if the number is 0 and -1 if the number is negative.
<b>SIN</b>	<b>SIN(number)</b>	Returns the sine of the given angle.
<b>SINH</b>	<b>Sinh(value)</b>	The <b>Sinh</b> function computes the hyperbolic sine of the argument.
<b>SKEW</b>	<b>SKEW(number1, number2, ...)</b>	Returns the skewness of a distribution. Skewness characterizes the degree of asymmetry of a distribution around its mean.
<b>SKEW.P</b>	<b>SKEW.P(number 1, [number 2],...)</b>	Returns the skewness of a distribution
<b>SLN</b>	<b>SLN(cost, salvage, life)</b>	Returns the straight-line depreciation of an asset for one period.
<b>SLOPE</b>	<b>SLOPE(known_y's, known_x's)</b>	Returns the slope of the linear regression line through data points in known_y's and known_x's. The slope is the rate of change along the regression line.
<b>SMALL</b>	<b>SMALL(array, k)</b>	Returns the k-th smallest value in a data set.
<b>SQRT</b>	<b>SQRT(number)</b>	Returns a positive square root.
<b>SQRTPI</b>	<b>SQRTPI(number)</b>	The SQRTPI function returns the square root of a given number multiplied by the mathematical constant, $\pi$ .
<b>STANDARDIZE</b>	<b>STANDARDIZE(x, mean, standard_dev)</b>	Returns a normalized value from a distribution characterized by mean and standard_dev.
<b>STDEV</b>	<b>STDEV(number1, number2, ...)</b>	Estimates the standard deviation based on a sample. The standard deviation is a measure of how widely values are dispersed from the average value (the mean).
<b>STDEV.P</b>	<b>STDEV.P(number1,[number2],...)</b>	The STDEV.P function calculates the standard deviation of a supplied set of values.

<b>STDEV.S</b>	<b>STDEV.s(number1,[number2],...)</b>	The STDEV.S function calculates the sample standard deviation of a supplied set of values.
<b>STDEVA</b>	<b>STDEVA(value1, value2 , ...)</b>	Estimates standard deviation based on a sample. The standard deviation is a measure of how widely values are dispersed from the average value (the mean). Text and logical values such as True and False are also included in the calculation.
<b>STDEVP</b>	<b>STDEVP(number1, number2, ...)</b>	Calculates standard deviation based on the entire population given as arguments.
<b>STDEVPA</b>	<b>STDEVPA(value1, value2, ...)</b>	Calculates the standard deviation based on the entire population given as arguments, including text and logical values.
<b>STEYX</b>	<b>STEYX(known_y's, known_x's)</b>	Returns the standard error of the predicted y-value for each x in the regression.
<b>SUBSTITUTE</b>	<b>SUBSTITUTE(text, old_text, new_text, instance_num)</b>	Substitutes new_text for old_text in a text string. Use SUBSTITUTE when you want to replace specific text in a text string; use REPLACE when you want to replace any text that occurs in a specific location in a text string.
<b>SUBTOTAL</b>	<b>SUBTOTAL (function_Number, ref1, (ref2)...) </b>	Subtotal function returns a subtotal in a list. Once the subtotal list is created, you can modify it by editing the Subtotal function.
<b>SUM</b>	<b>Sum( number1, number2, ... number_n )</b>	The <b>Sum</b> function adds all numbers in a range of cells and returns the result.
<b>SUMIF</b>	<b>SumIf( range, criteria, sum_range )</b>	<b>SumIf</b> function adds the specified range of cells by a given criteria.
<b>SUMPRODUCT</b>	<b>SUMPRODUCT(array1, array2, array3, ...)</b>	Multiplies corresponding components in the given arrays and returns the sum of those products.
<b>SUMSQ</b>	<b>SUMSQ(number1, number2, ...)</b>	Returns the sum of the squares of the arguments.
<b>SUMXMY2</b>	<b>SumXmY2( array1, array2 )</b>	The <b>SumXmY2</b> function calculates the sum of the squares of the differences between the corresponding items in the arrays and returns the sum as results.

<b>SUMX2MY2</b>	<b>SUMX2MY2(array_x, array_y)</b>	Returns the sum of the difference of squares of corresponding values in two arrays.
<b>SUMX2PY2</b>	<b>SUMX2PY2(array_x, array_y)</b>	Returns the sum of the sum of squares of corresponding values in two arrays. The sum of the sum of squares is a common term in many statistical calculations.
<b>SYD</b>	<b>SYD(cost, salvage, life, per)</b>	Returns the sum-of-years' digits depreciation of an asset for a specified period.
<b>T</b>	<b>T( value )</b>	The T function tests whether the given value is a text or not. When the given value is a text, then it returns the given text. Otherwise, the function returns as an empty text string.
<b>T.DIST</b>	<b>T.DIST(x,deg_freedom,tails)</b>	The T.Dist returns the Percentage Points (probability) for the Student t-distribution where a numeric value (x) is a calculated value of t for which the Percentage Points are to be computed.
<b>T.INV</b>	<b>T.INV(probability,deg_freedom)</b>	The T.INV returns the left-tailed inverse of the Student's t-distribution.
<b>TAN</b>	<b>TAN(number)</b>	Returns the tangent of a number.
<b>TANH</b>	<b>TANH(number)</b>	Returns the hyperbolic tangent of a number.
<b>TEXT</b>	<b>TEXT(value, format_text)</b>	Converts a value to text in a specific number format.
<b>TIME</b>	<b>TIME(hour, minute, second)</b>	Returns the decimal number for a particular time. The decimal number returned by TIME is a value ranging from 0 (zero) to 0.99999999, representing the times from 0:00:00 (12:00:00 A.M.) to 23:59:59 (11:59:59 P.M.).
<b>TIMEVALUE</b>	<b>TIMEVALUE(time_text)</b>	Returns the decimal number of the time represented by a text string. The decimal number is a value ranging from 0 (zero) to 0.99999999, representing the times from 0:00:00 (12:00:00 A.M.) to 23:59:59 (11:59:59 P.M.).

<b>TODAY</b>	<b>TODAY( )</b>	Returns the serial number of the current date. The serial number is the number of days since Jan 1, 1900.
<b>TRIM</b>	<b>Trim( text )</b>	The Trim function returns a text value with the leading and trailing spaces removed.
<b>TRIMMEAN</b>	<b>TRIMMEAN(array, percent)</b>	Returns the mean of the interior of a data set. TRIMMEAN calculates the mean taken by excluding a percentage of data points from the top and bottom tails of a data set.
<b>TRUE</b>	<b>True(stringvalue)</b>	The <b>True</b> function returns the logical value for True.
<b>TRUNC</b>	<b>TRUNC(number, num_digits)</b>	Truncates a number to an integer by removing the fractional part of the number.
<b>TYPE</b>	<b>TYPE( value )</b>	The TYPE function receives a value and returns an integer that represents the supplied value's data type.
<b>UNCHAR</b>	<b>UNICHAR(num)</b>	The Unichar function retrieves the Unicode character for a given numeric value.
<b>UNICODE</b>	<b>UNICODE(text)</b>	The UNICODE function calculates the number corresponding to the first character of the text.
<b>UPPER</b>	<b>Upper( text )</b>	The <b>Upper</b> function converts all characters in a text string to uppercase.
<b>VALUE</b>	<b>Value(range)</b>	The <b>Value</b> function computes the date or a string that contains the number, and converts it into number format.
<b>VAR</b>	<b>Var( number1, number2, ... number_n )</b>	The Var function returns the variance of a population based on sample of numbers.
<b>VARA</b>	<b>VarA( value1, value2, ... value_n )</b>	The <b>VarA</b> function returns the variance of a population based on a sample of numbers, text, and logical values (ie: TRUE or FALSE).
<b>VARP</b>	<b>VarP(listofvalues)</b>	The VarP function returns population variance of the listed values.
<b>VARPA</b>	<b>VARPA(value1, value2, ...)</b>	Calculates variance based on the entire population. In addition to numbers and text, logical values

		such as True and False are also included in the calculation.
<b>VDB</b>	<b>VDB(cost, salvage, life, start_period, end_period, factor, no_switch)</b>	Returns the depreciation of an asset for any period you specify, including partial periods, using the double-declining balance method or some other method you specify. VDB stands for variable declining balance.
<b>VLOOKUP</b>	<b>VLOOKUP(lookup_value, table_array, col_index_num, range_lookup)</b>	Searches for a value in the left most column of a table and then returns a value in the same row from a column you specify in the table. Use VLOOKUP instead of HLOOKUP when your comparison values are located in a column to the left of the data you want to find.
<b>WEBSERVICE</b>	<b>WEBSERVICE(url)</b>	The Webservice function returns data from a web service on the Internet or Intranet.
<b>WEEKDAY</b>	<b>WEEKDAY(serial_number,return_type)</b>	Returns the day of the week corresponding to a date. The day is given as an integer, ranging from 1 (Sunday) to 7 (Saturday) by default.
<b>WEEKNUM</b>	<b>WEEKNUM( serialNum, [returnType] )</b>	For a supplied a date, the WEEKNUM function returns an integer representing the week number (from 1 to 53) of the year.
<b>WEIBULL</b>	<b>WEIBULL(x,alpha,beta,cumulative)</b>	The <b>Weibull</b> function returns the Weibull distribution. This distribution is used in reliability analysis, such as calculating a device's mean time to failure.
<b>WEIBULL.DIST</b>	<b>WEIBULL.DIST(x,alpha,beta,cumulative)</b>	The Weibull.Dist function returns the Weibull Distribution.
<b>WORKDAY</b>	<b>WORKDAY(startDate, Days, [holidays])</b>	The Workday function returns a date that is a supplied number of working days (excluding weekends and holidays) ahead of a given start date.
<b>WORKDAY.INTL</b>	<b>WORKDAY.INTL (startDate, days, [weekend], [holidays])</b>	The WORKDAY.INTL function returns a date that is a supplied number of working days (excluding weekends and holidays) ahead of a given start date.
<b>XIRR</b>	<b>Xirr(cashflow, datelist, value)</b>	The <b>Xirr</b> function computes the internal rate of return for a schedule of possibly non-periodic cash flows.

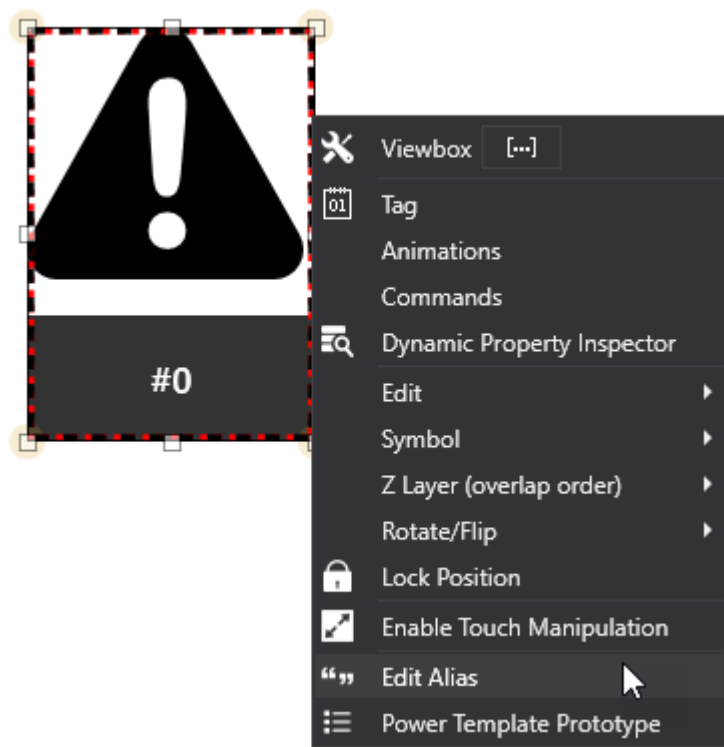
<b>XOR</b>	<b>XOR (logical_value1, logical_value2...)</b>	XOR function returns the exclusive OR for the given arguments.
<b>YEAR</b>	<b>YEAR(serial_number)</b>	Returns the year corresponding to a date. The year is returned as an integer in the range 1900-9999.
<b>YEARFRAC</b>	<b>YEARFRAC(start_date, end_date, [basis])</b>	YEARFRAC calculates the fraction of the year represented by the number of whole days between two dates (the start_date and the end_date). For instance, you can use YEARFRAC to identify the proportion of a whole year's benefits, or obligations to assign to a specific term.
<b>Z.TEST</b>	<b>Z.TEST(array,x,[sigma])</b>	Returns the one-tailed P-value of a z-test. For a given hypothesized population mean, x, Z.TEST returns the probability that the sample mean would be greater than the average of observations in the data set (array) — that is, the observed sample mean. To see how Z.TEST can be used in a formula to compute a two-tailed probability value, see the Remarks section below.
<b>ZTEST</b>	<b>ZTEST(array, u0, sigma)</b>	Returns the one-tailed probability-value of a z-test.

## 6.5. Aliases in Objects

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

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





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



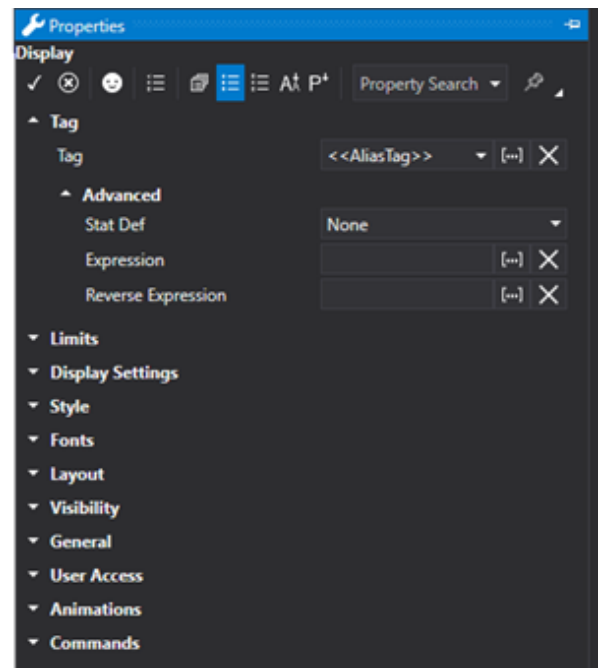
Aliases can only be connected to the project's Server Tags and not to those of other Servers.

## Alias Syntax

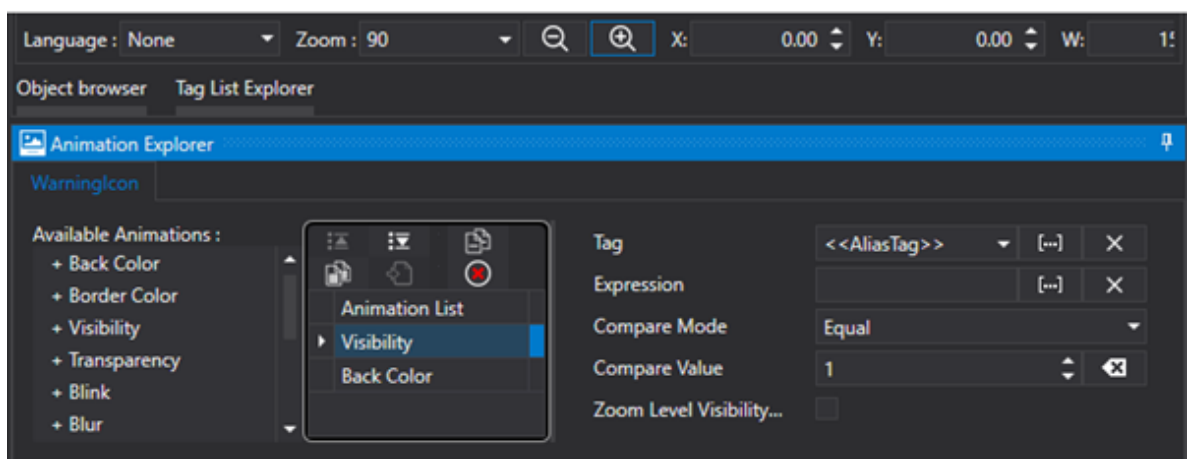
To be able to discriminate whether an Alias has been inserted in a field, it will need to respect a certain syntax. The Alias identification syntaxes should be characterized with the double angled brackets: `<<AliasName>>`.

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

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



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



The corresponding Alias Table would be:

Key	Value
AliasVariable	Folder1\ Variable1

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

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

## Alias Table

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

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

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

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

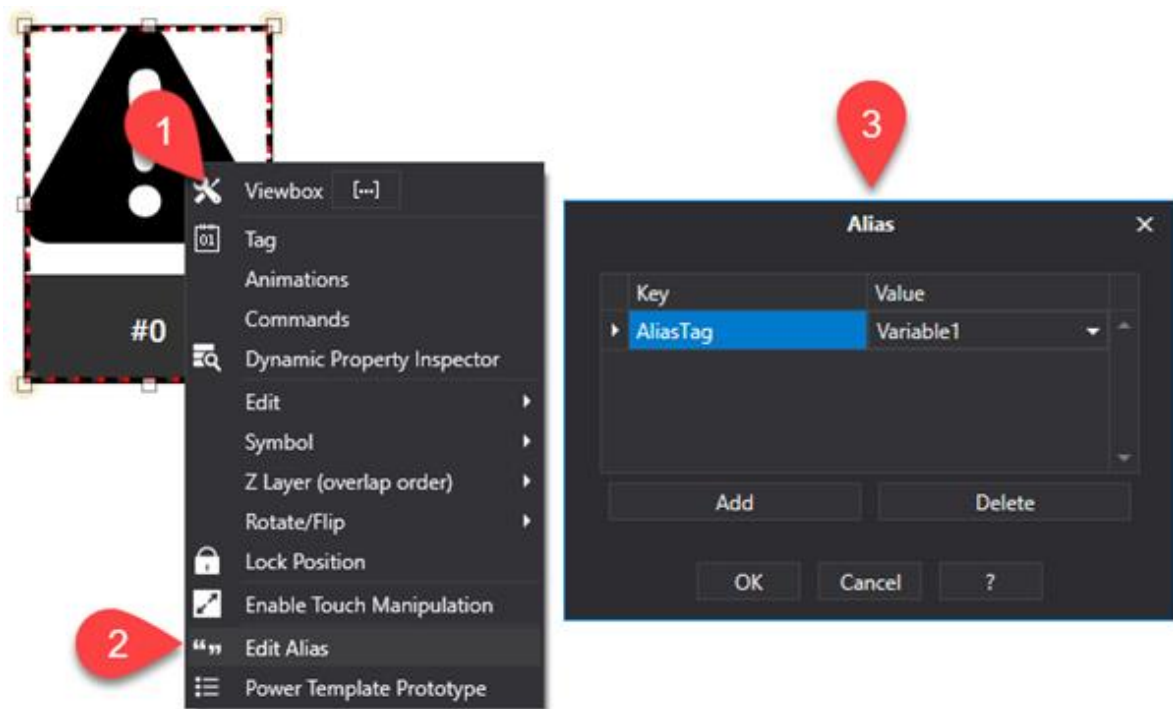
The Alias resolution is done starting from the lowest level to the highest level. For example, If an alias has been inserted in a component, its value will be searched for in that component's table. If the value is not found, it will then be searched for in the table of the parent symbol's containing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

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

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

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

The Alias Table will appear as below:



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

### Editing/Inserting Aliases

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

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

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

## Substituting Aliases at runtime

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

## Aliases in the Dynamic Property Inspector window

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

## Managing Aliases using Basic Script

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

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

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

In this case, the two functions execute the read and the write of the tag associated to the

<<AliasContextTag>> Alias in the Screen's Alias table.

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

## Managing Aliases in Dropping Code

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

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

```
'#Language "WWB.NET"

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

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

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

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

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

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

End Sub
```

## 6.6. Client system tags

Movicon provides users with a set of predefined tags that are used to obtain system information or for simulation curves. The System Tags are not published in the Server's address space and are only visible at Client level only.

The System Tags can be selected and assigned to project controls using the "Tag Browse Window".

Each System Tag has a specific function as described in the table below:

Variable Name	Data Type	Description
Blink2500ms	Integer	This variable blinks between "0" and "1" with a frequency of 2.5 seconds.
Blink5s	Integer	This variable blinks between "0" and "1" with a frequency of 5 seconds.
Blink10s	Integer	This variable blinks between "0" and "1" with a frequency of 10 seconds.
Blink30s	Integer	This variable blinks between "0" and "1" with a frequency of 30 seconds.
Blink200ms	Integer	This variable blinks between "0" and "1" with a frequency of 200 milliseconds.
Blink500ms	Integer	This variable blinks between "0" and "1" with a frequency of 500 milliseconds.
Blink1s	Integer	This variable blinks between "0" and "1" with a frequency of 1 second.
Sine	Float	This variable generates a simulation of a sine of an angle sine that varies from 0 to 360 degrees, with values ranging from "1" to "-1".
Square	Integer	This variable generates a simulation of a square wave with values ranging from "1" to "-1".
Triangle	Float	This variable generates a simulation of a triangle wave with values ranging from "1" to "-1".
Sawtooth	Float	This variable generates a simulation of a sawtooth wave with values ranging from "1" to "-1".
Pulse	Integer	This variable generates an impulse each second at value "1" to then return to value "0".
WhiteNoise	Float	This variable simulates a white noise at a value range between "1" and "-1".
GaussNoise	Float	This variable simulates a Gauss Noise at a value range between "1" and "-1".
DigitalNoise	Integer	This variable simulates a digital noise at a value range between "1" and "1".

LastHostNameUsed	String	This variable reports the name of the last Server to which the Client connected. The Client may also be connected to other Servers at the same time. This variable only shows the name of the Server subscribed to in the last connection.
CurrentTime	String	This variable reports the system's time.
CurrentDate	String	This variable reports the system's date.
CurrentLongDate	String	This variable reports the system's time and data. (e.g. Wednesday, January 27, 2016).
CurrentUser	String	This variable contains the name of the active user. If no user is logged on, the variable's value will be null.
MouseMove	Boolean	Indicates mouse is moving when active.
ActiveScreen	String	Name of the last active screen.
PI	Float	This variable reports the PC's performance index.
GPI	Float	This variable reports the PC's graphics performance index.
NLP	Integer	This variable reports the number of the PC's logic processors.
NP	Integer	This variable reports the number of the PC's physical processors.
CurrentRole	String	This variable contains the currently active user's role value. If no user is logged in, this value will be null.
CurrentAccessLevel	Integer	This variable contains the currently active user's access level value. If no user is logged in, this value will be '0'.
CurrentAccessMask	Integer	This variable contains the currently active user's access mask value. If no user is logged in, this value will be '0'.
ServerConnectionError	Integer	False = Client connected to the Server, True = Client connection error to Server. Server unavailable.
CurrentCulture	String	This variable reports the active language in the Client project.
CurrentConverter	String	This variable reports the Converter currently active in the Client project.
AlarmsSoundState	Boolean	This variable reflects the state of the Alarm sounds and is in both read and write. It is the same "SoundState" variable from the Server remapped in the Client.
AlarmsNumEnabled	Int64	Number of enabled Alarms. This is the same "NumEnabled" variable from the Server remapped in the Client.
AlarmsNumActiveOn	Int64	Number of active Alarms in the ON state. This is the same "NumActiveOn" variable from the Server remapped in the Client.

AlarmsNumActiveOff	Int64	Number of active Alarms in the OFF state. This is the same "NumActiveOff" variable from the Server remapped in the Client.
AlarmsNumActiveOnOff	Int64	Number of active Alarms in the ON and OFF status. This is the same "NumActiveOnOff" variable from the Server remapped in the Client.
AlarmsNumShelved	Int64	Number of active Alarms in the Shelve state. This is the same "NumShelved" variable from the Server remapped in the Client.
AlarmsNumNotAck Int64	Int64	Number of Alarms not yet Acknowledged. This is the same "NumNotAck" variable from the Server remapped in the Client.
MessagesNumEnabled	Int64	Number of enabled Messages. This is the same "NumEnabled" variable from the Server remapped in the Client.
MessagesNumActiveOn	Int64	Number of active Messages in the ON state. This is the same "NumActiveOn" variable from the Server remapped in the Client.
MessagesNumShelved	Int64	Number of Messages in the Shelve State. This is the same "NumShelved" variable from the Server remapped in the Client.
RedundancyActiveServerState	Int64	This variable indicates whether the local Redundancy Server is active or not. Each server reflects its own state. This is the same "RedundancyActiveServerState" variable from the Server remapped in the Client.
RedundancyActiveServerHostName	String	This variable reports the name of the Active Serve in the system, being the one in Control. This is the same "RedundancyActiveServerHostName" variable from the Server remapped in the Client.
RedundancyArrayAliveServerHostName	String Array	This variable reports the list of Server names currently started up in the system. This is the same "RedundancyArrayAliveServerHostName" variable from the Server remapped in the Client.



System variables are read only.

## 6.7. Styles

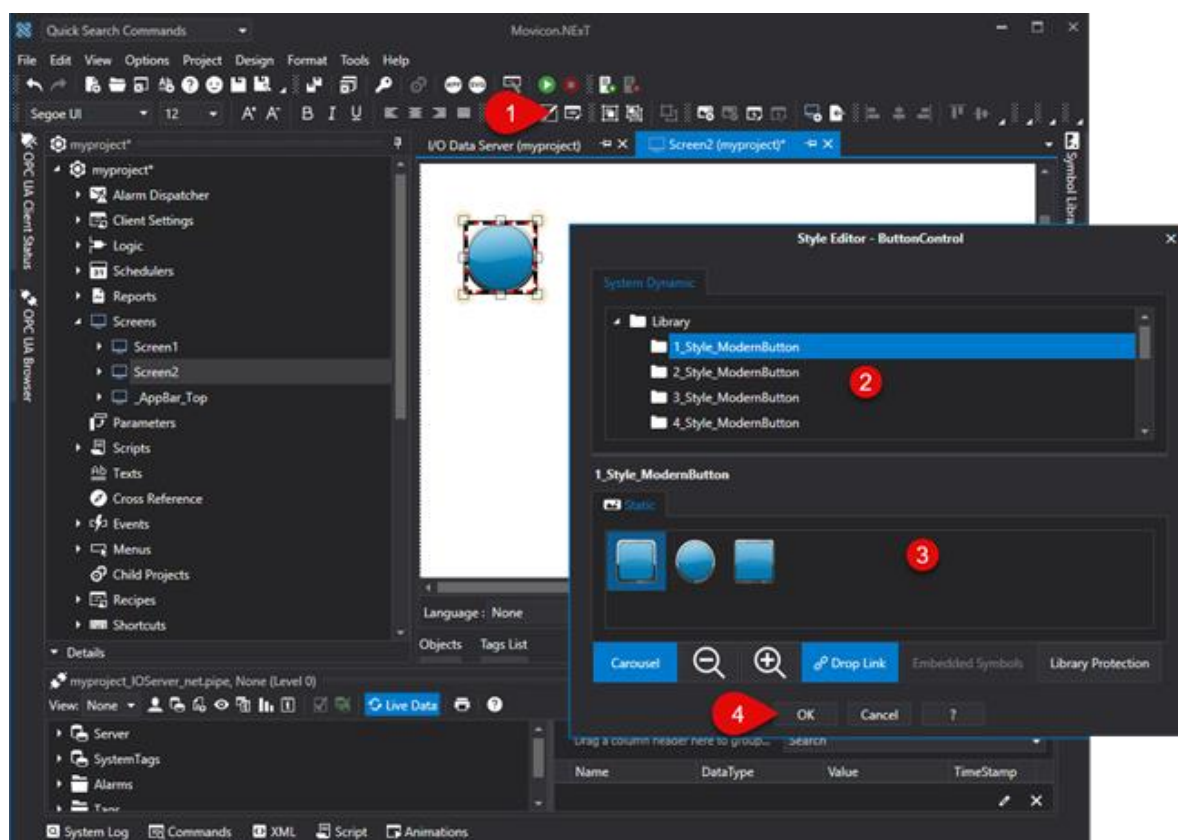
### 6.7.1. Using Styles

The Movicon.NExT graphics technology uses the intrinsic features of WPF to allow the uncoupling of each object's graphical part from its logic or mechanical functional part. This feature is expressed through the use of "Styles".

After having introduced a graphical symbol from the library, or an object from the toolbox, and configured it through its dynamic properties, it is possible to change its 'Style' at any given moment by using the Edit Style command from the Toolbar or from the menu Format > Style > Edit Style.



When using this command, a selection window will be displayed, that permits you to select the new graphics style to assign to the selected object.



By using this method the object will dynamically function in the same way as before but with a new look.



It is also possible to access the style properties through the object's command menu and selecting the "Edit Style" field.

## 6.7.2. Managing Styles in Movicon Next

Each XAML object can be styled using the 'Style' file. The Style file reference is defined in the XAML object's file. The style XAML object can also be inserted in the Movicon NeXT installation folder or added to the Objects in the Symbol Gallery.

In cases where you wish to use the styled XAML object in the Toolbox, you will need to perform these operations in order to do so:

1. Insert the XAML object's file in the C:\ProgramData\Progea\Movicon.NEXT\Toolbox\Screen folder.
2. Insert the style .xaml file in the "Resources " folder of the Movicon NeXT installation folder (for example: C:\ProgramData\Progea\Movicon.NEXT\Resources).

An example of a reference to the Style file reported in the XAML object's file is:

E.g.: `<TextBox Width="120" Height="35" Style="{DynamicResource TextBoxStyle1}" />`

where the style is defined "TextBoxStyle1" and the file with the style resource is inserted in C:\ProgramData\Progea\Movicon.NExT\Resources

In particular the "Macro-Categories" for which it is possible to manage the modification of the style through Movicon.Next with the appropriate format menu> style> edit style:

- Buttons
- CheckBoxes
- ProgressBars
- Sliders



For further info on editing styles in Movicon please refer to the chapter on "Using Styles"

The use of Styles in Movicon Next can be found in the Symbol Gallery's Digital and Analog tabs. The types of styles found in these windows are:

- **CheckBox** combined with the digital folder (e.g. objects that can change color)
- **ProgressBar** combined with the Analog folder (e.g. objects with filling animation)

The use of Styles in the Toolbox can be found in various types applied to all the object Macro-Categories:

- Buttons
- CheckBoxes
- ProgressBars
- Sliders



When applying styles to objects from the **Symbol Gallery**, Movicon will compare the structure of the C:\ProgramData\Progea\Movicon.NExT\Styles folder with the corresponding C:\ProgramData\Progea\Movicon.NExT\Symbols folder . If the sub folders in the corresponding paths have the same name, Movicon will use that style.

As regards to objects from the **"ToolBox"**, the style is applied by using a 'reference' system "reference" in the object's XAML.

For example: **File Reference** (Name Object.XAML.data).

Within the path we will have (FilePathObject base@FilePathStyle?StyleName):

C:\ProgramData\Progea\Movicon.NExT\Styles  
\Button.default@C:\ProgramData\Progea\Movicon.NExT\Styles\Button\Custom  
\EsStyleButton.xaml?EsButtonStyle1



At the moment only some categories enable you to edit their styles despite the fact that potentially all WPF objects should be edited in this way.,

## Add New Styles

It is therefore possible to add new style files directly to the MoviconNExT style installation folder C:\ProgramData\Progea\Movicon.NExT\Styles\ . While inserting a style, Movicon searches for .sgdefault

file corresponding to the folder in which the Style file resides. The styles defined in the Movicon library are:

- **Buttons** C:\ProgramData\Progea\Movicon.NExT\Styles\Button
- **CheckBoxes** C:\ProgramData\Progea\Movicon.NExT\Styles\CheckBox
- **ProgressBars** C:\ProgramData\Progea\Movicon.NExT\Styles\ProgressBar
- **Sliders** C:\ProgramData\Progea\Movicon.NExT\Styles\Slider

As regards to the ToolBox, the objects that refer to specific styles are characterized by these three files:

1. Image File to make object appear on the toolbox list (ObjectName.png)
2. Xaml File to define object (ObjectName.xaml) Eg: <TextBoxWidth="120"Height="35"Style="{DynamicResource TextBoxStyle1}" />
3. Reference File to reference the object's style (ObjectName.xaml.data)
  - FilePathObject base@FilePathStyle?StileName
  - Eg.:C:\ProgramData\Progea\Movicon.NExT\Styles\Button.default@C:\ProgramData\Progea\Movicon.NExT\Styles\Button\Custom\EsStyleButton.xaml?EsButtonStyle1

### Rules for Creating new Styles

There are a variety of editors that can be used to create Style files starting with the most complex (e.g. Blend), to the less performing ones (e.g. Note pad). However, no matter which one is used it is important that you create the style according to the rules below in order to be supported in MoviconNExT.

#### The styles must be included in a ResourceDictionary:

```
<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008" mc:Ignorable="d"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"> </ResourceDictionary>
```



For further details about creating styles please refer to the Microsoft guide.

A simple example of creating a style would be:

```
<Style x:Key="EsButtonStyle2" TargetType="{x:Type Button}">
<Setter Property="Template">
    <Setter.Value>
        <ControlTemplate TargetType="{x:Type Button}">
            <Grid x:Name="grid">
                <Viewbox Stretch="Fill">
                    <Grid x:Name="modernButton1_tond" Height="90.704" Width="90.708">
                        <Rectangle Tag="Background" x:Name="Back"
Fill="Red" Opacity="1"/>
                    </Grid>
                </Viewbox>
                <TextBlock Tag="Title" MaxWidth="70" TextWrapping="Wrap"
VerticalAlignment="{Binding VerticalContentAlignment, RelativeSource={RelativeSource
TemplatedParent}}"
HorizontalAlignment="{Binding HorizontalContentAlignment, RelativeSource={RelativeSource
TemplatedParent}}"
Text="{Binding Content, RelativeSource={RelativeSource TemplatedParent}}" d:IsHidden="True"/>
            </Grid>
        </ControlTemplate>
```

```

        </Setter.Value>
    </Setter>
</Style>

```



The Tag = "Background" is used to mark the objects to which the background will be applied, if specifically modified from the menu format> style> color (Edit Back Color). The TextBlock can be used to display a text above the control that can then be modified in design, from the CommonProperties window.

- **Keep lines of code to be inserted to a minimum in order for Movicon to load quickly.** The objects with complex Paths should be optimized (some editors, such as Blend, allow complex paths to be grouped together by using the 'combinepath' command).
- **In order to discriminate style behaviour in cases where the project is being run on a low performing system (such as the WebClient), you can define style behaviours using Triggers instead of VisualStates.** VisualStates establish the object's behaviour at a determined state without offering the possibility to discriminate values of those properties exposed. Alternatively, Triggers offer the possibility to activate determined StoryBoards only when certain conditions occur. Movicon provides a DependencyProperty type that determines whether the object has been loaded in a low performing system or not. The exact property used for this is called **ScreenDocument.RunningOnSlowPCMachine**. In order to reference this property you will need to add the "NameSpace" to the ResourceDictionary reference:

```
xmlns:mynsp="clr-namespace:ScreenSettings;assembly=ScreenSettings"
```

Once the ScreenSettings library has been referenced, the trigger can be used in the following way:

```

<Condition Property="mynsp:ScreenDocument.RunningOnSlowPCMachine" Value="False"/>

<Condition Property="mynsp:ScreenDocument.RunningOnSlowPCMachine" Value="True"/>

```

The storyboards to be activated in the Triggers can be defined within the style file's resource session.

#### Example:

```

<Style x:Key="EsButtonStyle3" TargetType="{x:Type Button}">
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="{x:Type Button}">
                <ControlTemplate.Resources>
                    <Storyboard x:Key="MouseOver">
                        <DoubleAnimationUsingKeyFrames Storyboard.TargetProperty="(UIElement.Opacity)"
                            Storyboard.TargetName="Back">
                            <EasingDoubleKeyFrame KeyTime="0" Value="0.75"/>
                        </DoubleAnimationUsingKeyFrames>
                    </Storyboard>
                    <Storyboard x:Key="NotMouseOver">
                        <DoubleAnimationUsingKeyFrames Storyboard.TargetProperty="(UIElement.Opacity)"
                            Storyboard.TargetName="Back">
                            <EasingDoubleKeyFrame KeyTime="0" Value="1"/>
                        </DoubleAnimationUsingKeyFrames>
                    </Storyboard>
                </ControlTemplate.Resources>
                <ControlTemplate.Triggers>
                    <MultiTrigger>

```

```

    <MultiTrigger.Conditions>
        <Condition Property="IsMouseOver" Value="True"/>
    </MultiTrigger.Conditions>
    <MultiTrigger.EnterActions>
        <BeginStoryboard x:Name="MouseOverStoryboard" Storyboard="{StaticResource
        MouseOver}"/>
    </MultiTrigger.EnterActions>
    <MultiTrigger.ExitActions>
        <StopStoryboard BeginStoryboardName="MouseOverStoryboard"/>
        <BeginStoryboard Storyboard="{StaticResource NotMouseOver}"/>
    </MultiTrigger.ExitActions>
</MultiTrigger>
</ControlTemplate.Triggers>
    <Grid x:Name="grid">
        <Viewbox x:Name="viewbox" Stretch="Fill">
            <Viewbox.Effect>
                <DropShadowEffect Opacity="0"/>
            </Viewbox.Effect>
            <Grid x:Name="modernButton1_tond" Height="90.704" Width="90.708">
                <Rectangle Tag="Background" x:Name="Back"
Fill="{DynamicResource E_G_Arcs}" Opacity="1"/>
            </Grid>
        </Viewbox>
        <TextBlock Tag="Title" MaxWidth="70" TextWrapping="Wrap"
VerticalAlignment="{Binding VerticalContentAlignment, RelativeSource={RelativeSource
TemplatedParent}}"
HorizontalAlignment="{Binding HorizontalContentAlignment, RelativeSource={RelativeSource
TemplatedParent}}"
Text="{Binding Content, RelativeSource={RelativeSource TemplatedParent}}" d:IsHidden="True"/>
    </Grid>
</ControlTemplate>
</Setter.Value>
</Setter>
</Style>

```

- **It is good to use the colors (Brush) dynamically**, without defining them statically within the control, A very simple way can be to use the colors that are already loaded by Movicon.NExT. The list and names of colors can be found in the Brush setting window, accessible from the Format> style> color menu for setting the BackColor.

To use these colors simply use this command:

{DynamicResource COLOR\_NAME} (eg.: {DynamicResource E\_G\_Arcs}).

You can also add new colors to the resource file loaded by MoviconNExT in the C:\ProgramData\Progea\Movicon.NExT\Resources folder:

- LinearGradient (LinearGradientBrush)
- RadialGradient (RadialGradientBrush)
- SolidColor (SolidColorBrush)
- VisualBrush (VisualBrush)
- FillingBrush (DrawingBrush)



Changing the colors in this folder will change the aspect of the entire symbol library.

- **Ultimately, it is always best to use unique keys for Style definitions and Resources such as the Brushed.** The best method to use is the Guid (**G**lobally **U**nique **I**dentifier) as a key. Different tool types can be used for creating GUIDs that permit unique codes to be generated randomly (based on data such as date and time of the machine with which the code is generated).

## Managing styles in Movicon.Next with external files

The Font style of the entire application or just some of its screens can be defined using an appropriate file as described below:

### The application's general style file

When creating a **default.style** file in the application's Documents folder (for further details see the example below) you will be able to edit the valid style definitions for the whole application.

### Specific screen style file

When creating a **<screenname>.xaml.style** in the **Screen** folder, you will be able to edit the valid style definitions just for the <screenname> screen only. Style files that are created in this folder will be loaded into the general style folder and their settings will have priority over those in the file default.style file.

**Style file syntax:** the style file syntax is the one used for each xaml resource file.

For example:

```
<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
```

```
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
```

```
<Style TargetType="{x:Type TextBox}" x:Key="MyTextBoxStyle">
```

```
    <Setter Property="FontFamily" Value="Constantia"/>
```

```
    <Setter Property="FontWeight" Value="Bold"/>
```

```
    <Setter Property="FontStyle" Value="Italic"/>
```

```
    <Setter Property="FontSize" Value="48"/>
```

```
<Setter Property="Foreground" Value="#FFBDEC12"/>
```

```
</Style>
```

```
</ResourceDictionary>
```

## Associating styles to controls

To associate controls with custom styles, simply insert the style reference directly in each control to consequently create a custom library.

However the custom library with the specific control styles defined in one project will not be recognized by subsequent new projects when run. For example, if you define a specific style in a 'Custom' Toolbox control in one project and then later create a new project using the same control, this control will function with its default style when the new project is run and not the one it was customized with. This is because the new project, in which the control is being used in runtime, is not the one in which the custom library containing the specific style was created in. Therefore associated custom control styles are specific to the project in which they were created and cannot be used automatically by other projects.

In this phase the steps to use the style defined in the files are as follows:

1. Create a folder in the toolbox for the controls to be associated with styles (e.g.: C:\ProgramData\Progea\Movicon.NExT.3.1\Toolbox\Screen\CustomControls)
2. Copy the controls to be styled
3. Edit the controls by adding the style reference (eg. Style="{DynamicResource MyAlarmWindowStyle}")
4. Create the style file to be applied, in the default.style file for example, with the '<screennamescreen>.xml.style' name.

An example of a style definition for the **TextBox**:

### Style File

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<ResourceDictionary
```

```
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
```

```
xmlns:s="clr-
```

```
namespace:Microsoft.Surface.Presentation.Controls;assembly=Microsoft.Surface.Presentation"
```

```
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
```

```
<Style TargetType="{x:Type TextBox}" x:Key="MyTextBoxStyle">
```

```

<Setter Property="FontFamily" Value="Gill Sans" />

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="FontStyle" Value="Italic" />

<Setter Property="FontSize" Value="10" />

<Setter Property="Foreground" Value="#FFBDEC12" />

<Setter Property="Background" Value="#FFFAEBD7" />

</Style>

</ResourceDictionary>

```

## Text Control

```

<?xml version="1.0" encoding="utf-8"?>

<TextBox xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Width="100" Height="30" MinWidth="5"
MinHeight="5" BorderThickness="0" Background="{x:Null}" IsReadOnly="True" IsHitTestVisible="False"
Text="Text" Style="{DynamicResource MyTextBoxStyle}" />

```

Example of a style definition for the **AlarmWindow**:

## Style File

```

<?xml version="1.0" encoding="utf-8"?>

<ResourceDictionary

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:s="clr-
namespace:Microsoft.Surface.Presentation.Controls;assembly=Microsoft.Surface.Presentation"

xmlns:aw="clr-namespace:AlarmWindow;assembly=AlarmWindow"

xmlns:converters="clr-namespace:Converters;assembly=Converters"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">

    <converters:FontSettings x:Key="rowFontSettings">

```



```
<converters:FontSettings.FontFamily>

  <FontFamily>Gill Sans</FontFamily>

</converters:FontSettings.FontFamily>

<converters:FontSettings.FontSize>

  10

</converters:FontSettings.FontSize>

<converters:FontSettings.FontStyle>

  Normal

</converters:FontSettings.FontStyle>

<converters:FontSettings.FontWeight>

  Normal

</converters:FontSettings.FontWeight>

</converters:FontSettings>

<converters:FontSettings x:Key="headerFontSettings">

  <converters:FontSettings.FontFamily>

    <FontFamily>SketchFlow Print</FontFamily>

  </converters:FontSettings.FontFamily>

  <converters:FontSettings.FontSize>

    25

  </converters:FontSettings.FontSize>

  <converters:FontSettings.FontStyle>

    Normal

  </converters:FontSettings.FontStyle>

  <converters:FontSettings.FontWeight>

    Normal

  </converters:FontSettings.FontWeight>

</converters:FontSettings>
```

```

<Style TargetType="{x:Type aw:GridAlarmWindow}" x:Key="MyAlarmWindowStyle">

    <Setter Property="AlarmAreaFontSettings" Value="{StaticResource rowFontSettings}" />

    <Setter Property="HeaderFontSettings" Value="{StaticResource headerFontSettings}" />

</Style>

</ResourceDictionary>

```

## AlarmWindowControl

```

<?xml version="1.0" encoding="utf-8"?>

<aw:GridAlarmWindow xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:aw="clr-
namespace:AlarmWindow;assembly=AlarmWindow" Width="600" Height="300"
Tag="ProblematicXamlWriter" Style="{DynamicResource MyAlarmWindowStyle}">

    <aw:GridAlarmWindow.Resources>

        <Style TargetType="{x:Type ListView}" x:Key="AlarmDataSource"> </Style>

    </aw:GridAlarmWindow.Resources>

</aw:GridAlarmWindow>

```



In order to setup references to controls in the style definition phase, they must be inserted in the controls' name spaces. For example, the alarm window name space is:

```
xmlns:aw="clr-namespace:AlarmWindow;assembly=AlarmWindow"
```

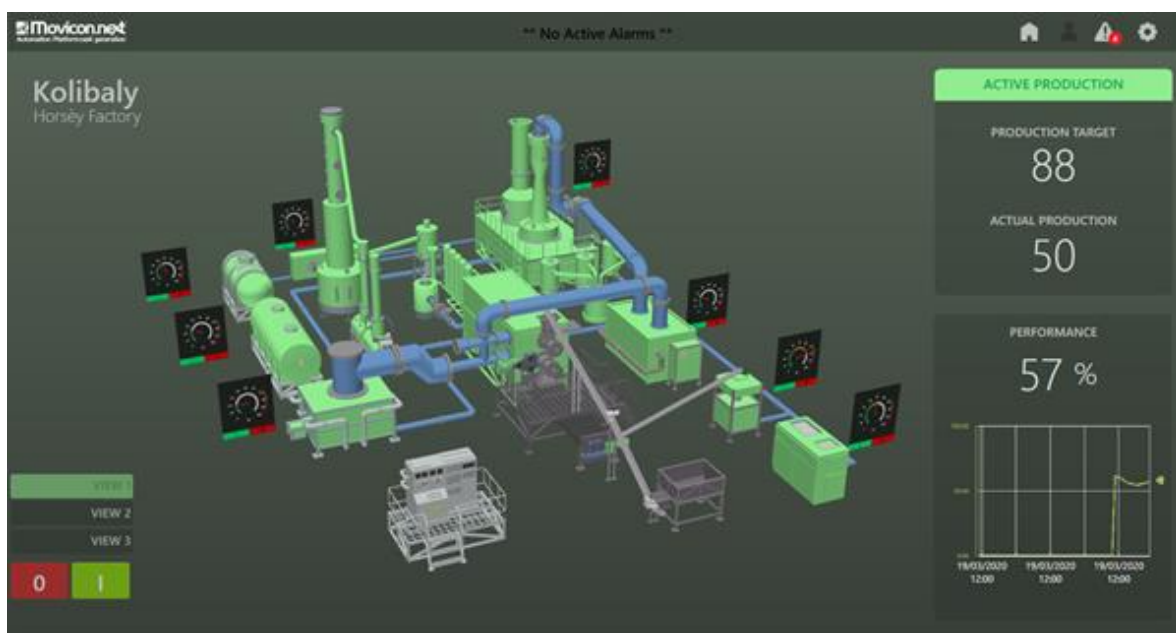
Once this has been done, the target type in the style will be:

```
TargetType="{x:Type aw:GridAlarmWindow}"
```

## 6.8. 3D graphics

### 6.8.1. 3D Graphics

Movicon.NExT uses XAML graphics with WPF technology which enables you to insert and use objects with 3D graphics. Each individual 3D symbol component of the object can be animated and the object's observation angle can also be varied (camera).



This diagram illustrates the concepts used for managing 3D objects within the Movicon.NEXT screens. The system accepts previously created 3D models that have been made available for importing into Movicon screens with the aim to render the screens dynamic and interactive with the visualization and supervision system users.

Once a 3D model has been inserted, it can be associated with project tags in order to manage the dynamic animation functions. These functions can be associated to each element or group of elements configured within the inserted 3D model scene.

This will allow users to create powerful animated graphical interfaces in 3D with objects that can, for instance, change colour according to the associated tag status or move around within the scene in function with the tag values.

In addition to this, the operator can interact with the 3D object on screen and rotate or zoom it as pleased. Different 'views' can be stored and recalled for later use. The views may be called on event so that the object can be positioned appropriately in function with the system's status or events.



Movicon.NEXT is not yet designed for editing or creating 3D scenes and graphics models. Movicon's job is to insert this objects on screen and provide real-time animation graphics by linking dynamic Tags to animation types. The Movicon symbol library contains some example 3D graphics models allowing the user use their own 3D models (generally created from Cad 3D or simulation software) by importing and inserting them on screen.



An XAML Viewer can be used, such as the Viewer3DS di AB4D to verify whether the XAML file containing the 3D model can be imported and displayed correctly in Movicon.NEXT.



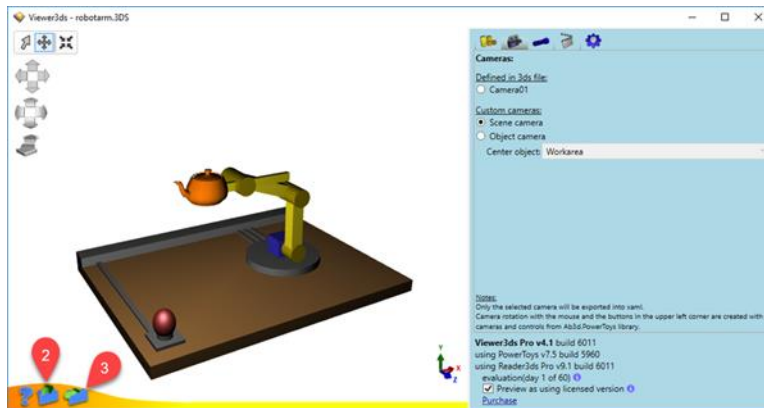
The 3D graphics management is not supported during Web Client visualization.

## Converting 3D files to XAML

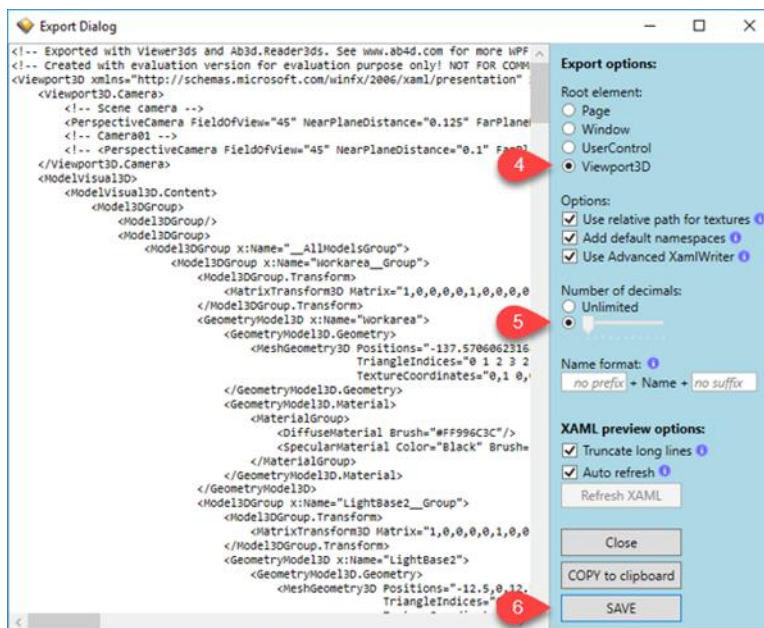
Movicon.NExT has not yet been designed to change or create 3D graphics models directly within the development environment. Therefore, 3D graphics will need to be converted into XAML before importing them.

This simplest way to do this is to:

1. Convert the 3D file to ".3DS" (Autodesk 3DS Max).
2. By means of using the Viewer3DS di AB4D software, open the ".3DS" file.
3. Select the XAML format for exporting file.



4. Make sure you select the "Viewport3D" option.
5. Lower the number of "Decimals" from Unlimited to Zero.
6. At this point you will be able to save the file in XAML format.



## XAML File Structure for importing

If you wish to animate the elements of an imported 3D symbol, you will need to take precautions importing the ".xaml" file beforehand. This is done by adding the following tag in the initial code:

Tag = "ProblematicXamlWriter"

For example, if the original file has this structure:

```
<Viewbox x: Name = "MOV3DViewbox" ClipToBounds = "true" xmlns =  
"http://schemas.microsoft.com/winfx/2006/xaml/presentation"
```

```
xmlns: x = "http://schemas.microsoft.com/winfx/2006/xaml" xmlns: d =  
"http://schemas.microsoft.com/expression/interactivedesigner/2006"  
xmlns: c = "http://schemas.openxmlformats.org/markup-compatibility/2006" c: Ignorable = "d">
```

.....

Simply add the tag as follows:

```
<Viewbox x: Name = "MOV3DViewbox" ClipToBounds = "true" xmlns =  
"http://schemas.microsoft.com/winfx/2006/xaml/presentation"  
xmlns: x = "http://schemas.microsoft.com/winfx/2006/xaml" Tag = "ProblematicXamlWriter"  
xmlns: d = "http://schemas.microsoft.com/expression/interactivedesigner/2006" xmlns: c =  
"http://schemas.openxmlformats.org/markup-compatibility/2006"  
c: Ignorable = "d">
```

.....

This precaution will ensure that the object's 3D animation functions correctly when imported by means of using the "Import XAML" command and inserted on screen from the Symbol Library using the "merge code" mode (incorporates code).

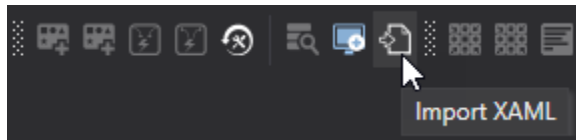
This ".xaml" file modification can be done with any text editor.

The WPF 3D technology offers the advantage of being able to use the graphics acceleration hardware to the full, if available. As an alternative, WPF also provides fully automatic software rendering features that greatly simplify 3D management for the user without having to revert to using expensive hardware configurations or setups.

## Importing 3D files on screen

By using the "**Import XAML**" command available from the "**Insert**" group Toolbar, select the desired graphics file and import it into your project to create a new Movicon.NExT symbol.

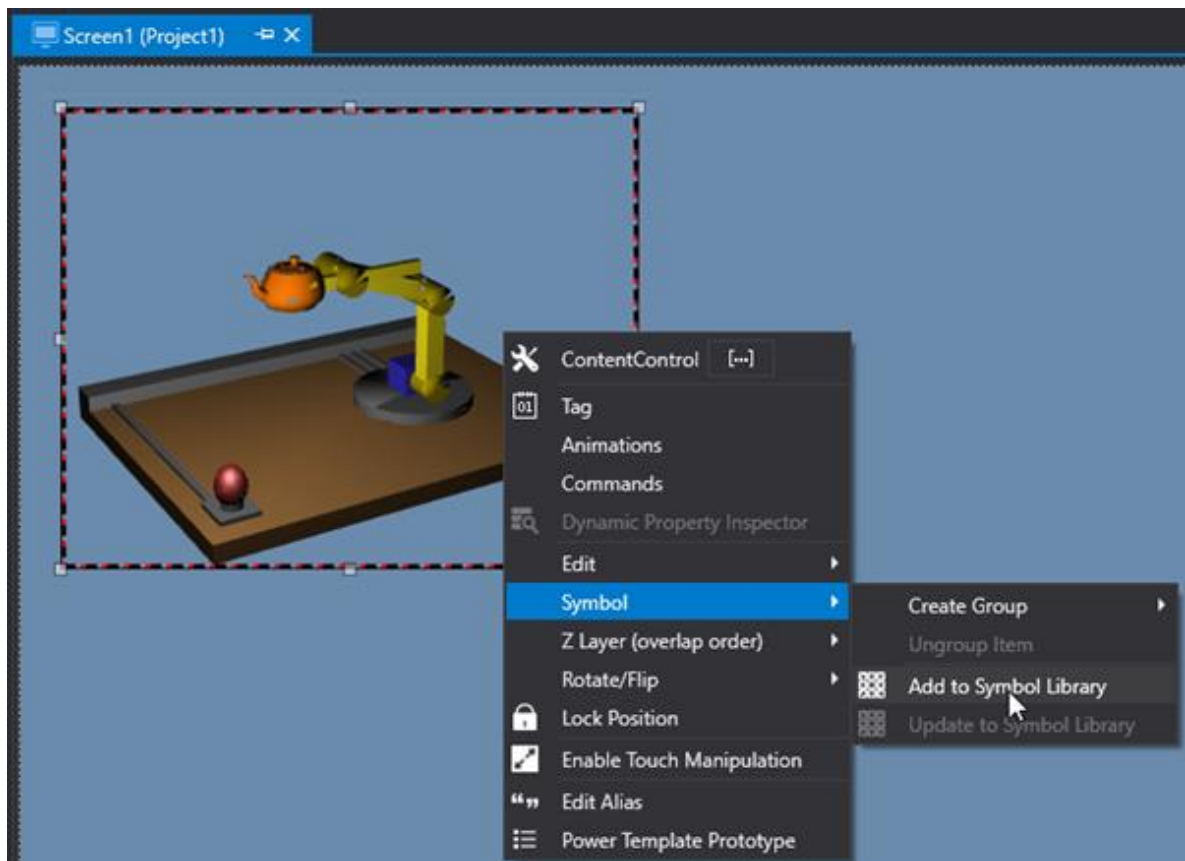
Once the object has been imported on screen, select it and group it using the CTRL+G command and then insert it in the project's customized symbol library.



From the file's browser window, make sure that you specify the file format ((filter selection).



Attention! If you need to animate elements of an imported 3D file, you will need to insert the "ProblematicXamlWriter" tag in the symbol's code as indicated in the above paragraph on "Xaml file structure importing".



## 3D elements in the Symbol Library

The Movicon Symbol Library contains some example 3D models that can be inserted in screens and used as needed. The user can also import their own 3D models as described previously, and save them in the symbol library to create their own custom 3D libraries by using the techniques described in the "Movicon Symbols" topic to create them.

However, it is also possible to insert the ".xaml" file of the 3D object directly into the library folder for immediate use. For example by arranging of the "3DName.xaml" file of the object you want to use, you can import the file directly by using the "Import XAML" command described above. This file can also be copied directly into the symbol library (for example in the path ".. \ Project1 \ Project1 \ Symbols \ 3DObjects" in order to insert the object in the project library).

What ever mode you use, the 3D object can be made available in the library and inserted on screen using the "drop link" mode (drag connection) that "merge code" (incorporates the code).

### 6.8.2. 3D Animation Graphics

After you have inserted a 3D model on screen, you can then set it with dynamic animation which is done by associating the relevant tags.

The dynamic animation functions can be associated to each individual component of the 3D model, or groups of components. **Group definitions must be predefined in the model**

**beforehand.** Movicon.NExT will only select the group and dynamically animate all the components belonging to it.

## Selecting elements or groups in 3D models

To proceed with selecting a 3D model element you must first enable the "**Enable 3D Edit**" command from the object's command menu. After having enabled the object's edit mode, click on an element to select it or use SHIFT+Click to select a group of elements (predefined in the model beforehand). Elements will highlight when selected.

## Animation property settings



For further information on "Graphic Animations" please refer to the relating topic.

### 3D X Angle Rotation

The object's angle rotation on the X axis in function with the Tag value.

- **Target Angle:** Sets the rotation value for the selected angle.
- **Center X:** Sets the rotation center point on the X axis.
- **Center Y:** Sets the rotation center point on the Y axis
- **Center Z:** Sets the rotation center point on the Z axis.

### 3D Y Rotation Angle

The object's angle rotation on the Y axis in function with the Tag value.

- **Target Angle:** Sets the rotation value for the selected angle.
- **Center X:** Sets the rotation center point on the X axis.
- **Center Y:** Sets the rotation center point on the Y axis
- **Center Z:** Sets the rotation center point on the Z axis.

### 3D Z Rotation Angle

The object's angle rotation on the Z axis in function with the Tag value.

- **Target Angle:** Sets the rotation value for the selected angle.
- **Center X:** Sets the rotation center point on the X axis.
- **Center Y:** Sets the rotation center point on the Y axis
- **Center Z:** Sets the rotation center point on the Z axis.

### Back Color

Back Color based on the Tag's threshold values. Please refer to the relating topic: "Back Color Animation"

### 3D Move X

The object's linear movement on the X axis in function with the Tag value.

- **Move X:** sets the number in pixels with which the object will be moved along the X axis.

### 3D Move Y

The object's linear movement on the Y axis in function with the Tag value.

- **Move Y:** sets the number in pixels with which the object will be moved along the Y axis.

### 3D Move Z

The object's linear movement on the Z axis in function with the Tag value.

- **Move Z:** sets the number in pixels with which the object will be moved along the Z axis.

### Scale 3D

Object's scale size in function with the Tag value.



- **Object Scale:** Sets the scale factor to reach where 1 refers to the object's standard size.
- **Center X:** Sets the center of the scale factor on the X axis.
- **Center Y:** Sets the center of the scale factor on the Y axis.
- **Center Z:** Sets the center of the scale factor on the Z axis.

## Transparency

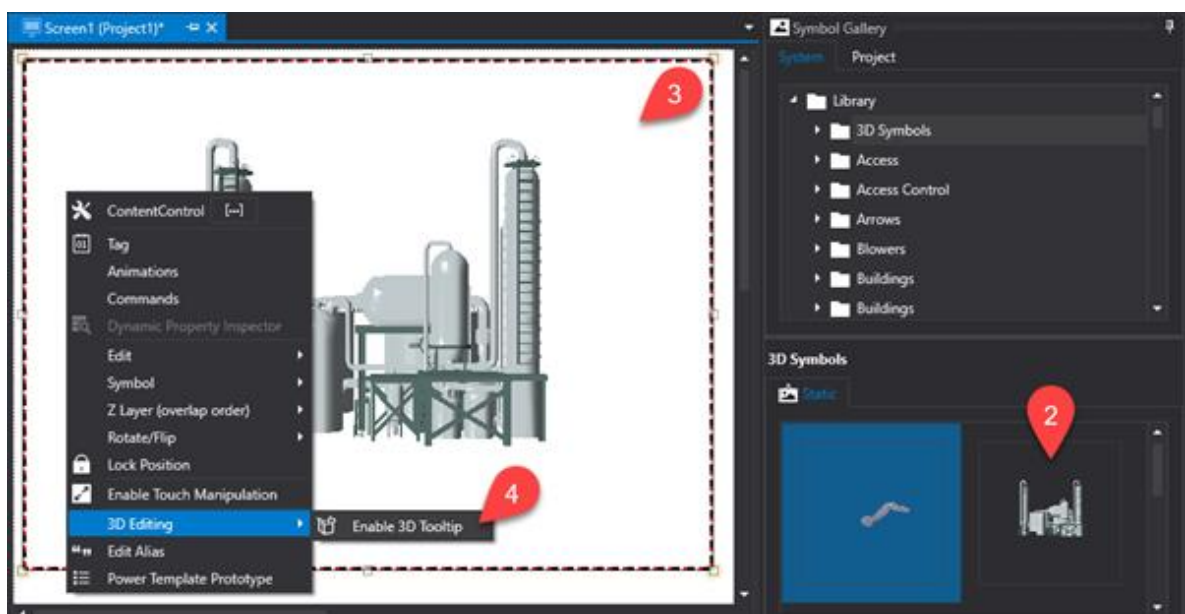
The Transparency animation of a 3D object manages the transparency but not the animation time which is always managed as if set at zero.

- **Transparency:** Sets the transparency factor to reach where 1 refers to full transparency and 0 corresponds to invisible object.
- **Visible On 3D Camera Zoom:** This property is used for managing the animation with a 3D camera zoom level instead of the associated tag value, to make the object visible or invisible. The value inserted in the "Transparency" property will in this case be used as the zoom factor that when exceeded will make the object invisible. However, please consider that the scale transformation will not be applied to the object but to the camera, therefore the result will be the opposite. This means that when the object is smaller, the scale value is bigger. If "0.5" is inserted as the Transparency value, the object will remain invisible until enlarged by 150%. Conversely, when set with "1.5" the object will become invisible when reduced by 150%.

## 3D Animation example

We shall now look at the basic techniques used for applying graphical animation to objects. To do this we will use the "NITRO ENGINE" object from the symbol library as an example to animate the engine's transmission coupling upon Start-Stop command activation:

1. Open an existing screen or add a new one.
2. Open the Symbol Gallery and use the mouse click to select the "Refinery" object from the "3D Symbols" group.
3. Position the mouse pointer on an area within the screen and click to release and insert the object. Size and position the object as pleased.
4. Select the object and access the commands menu by clicking the icon located on the object's borders and select and activate the "**3D Edit**" item.



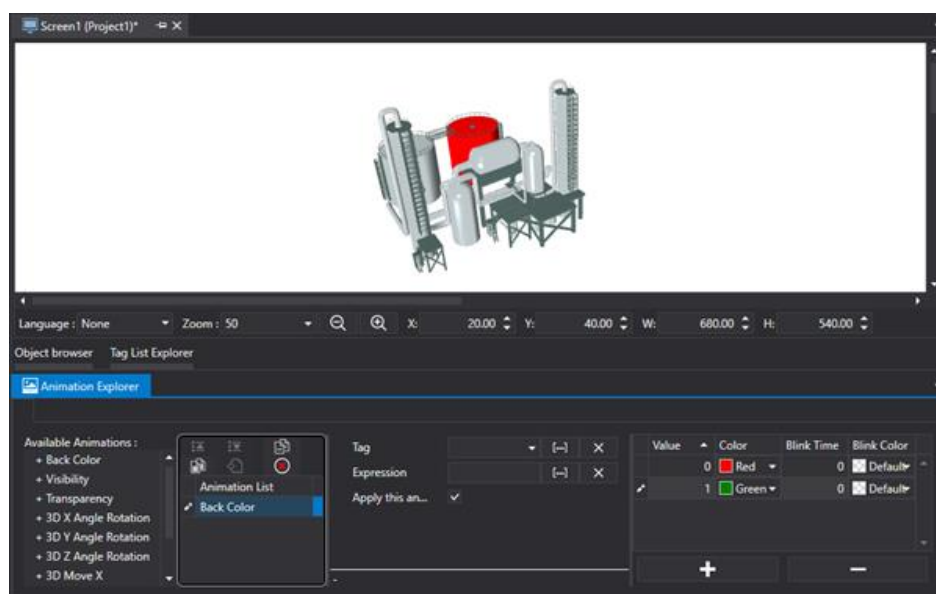
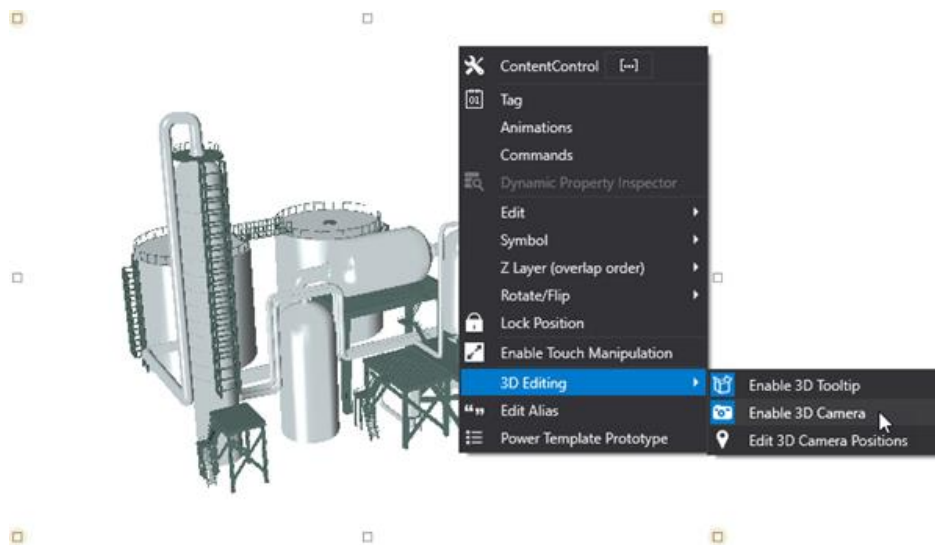


The "3D Edit" button is available from the Object's Commands Tooltip

- Note that when clicking the various 3D symbol components, the selected symbol's parts will highlight with a pulsating color effect to show which parts of the 3D symbol the animations refer to.

*When the 3D symbol has been created as a component of a "Group", the entire group will be selectable with the SHIFT+Click command.*

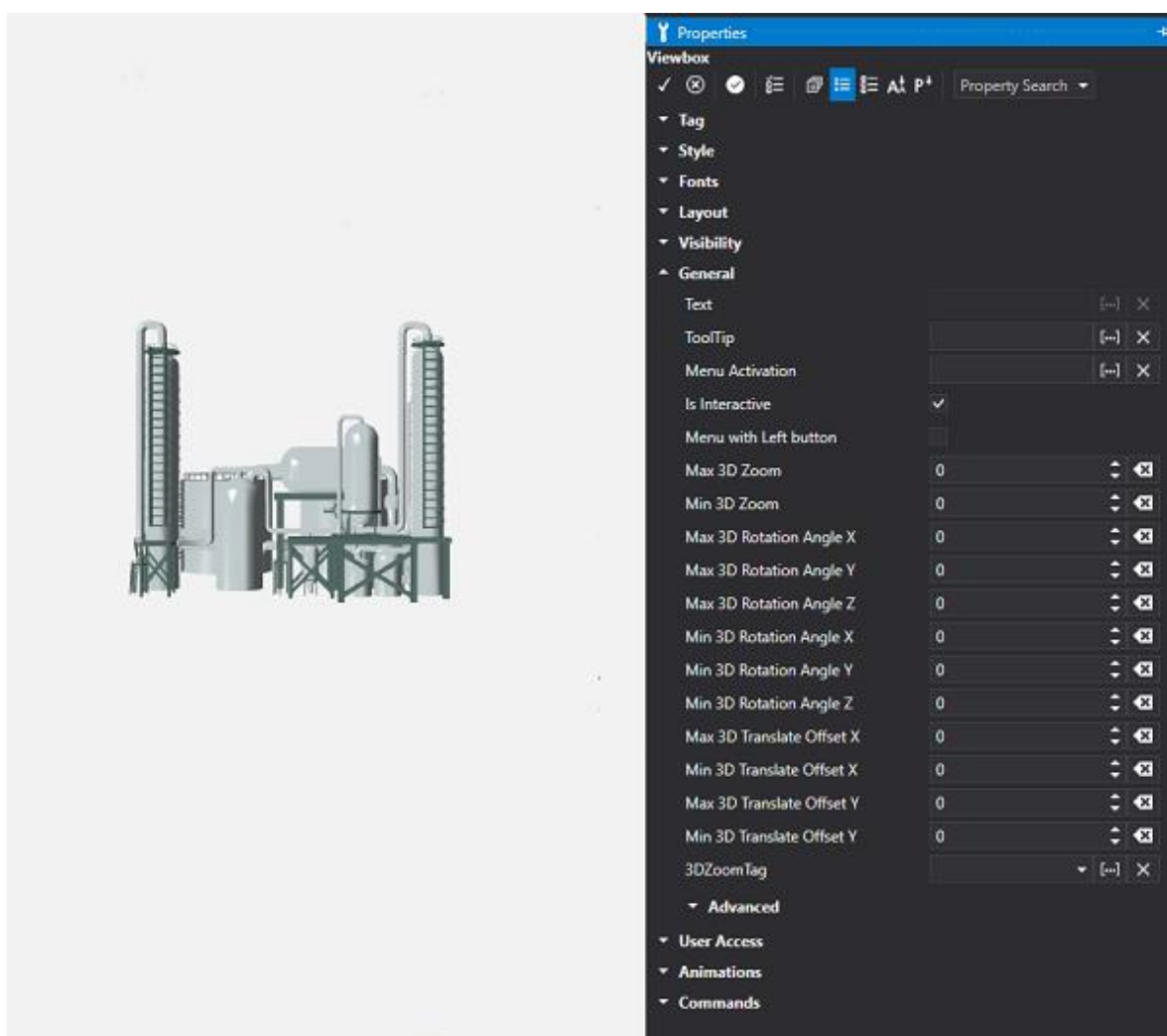
- The 3D object can also be rotated while being edited to add other components to it or to record different events viewed (different 3D camera scene positions), which can then be set by the user in runtime with a right mouse click.
  - To use the rotation and recording functions of eventual views, use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



6. After having selected the 3D component model desired for animating, open the object's command menu again and use the "**Common Property Editor**" command.
7. Select the "**Animation**" tab from the dialog window that opens to access the **graphical animation commands** available for individual components and component groups that compose the 3D model and which are listed in the table below.
8. In our example we have used the animated object's Back Color. Any other type of graphical animation can be applied by using the same procedures.
10. Go to the "Animation" window and select the desired dynamic animation command. In this example we have selected the "Back Color" animation.
11. When inserting the "Back Color" animation on the list of animations for the selected component, a window will appear on the right to define its settings. In this window it will be possible to insert the colour threshold values by adding the colour thresholds desired to function with the values that will be obtained by the tag (e.g. 0 = the color red, 1 = the color green). It will then be possible to select the Tag to associate as well as other parameters as already seen with the other dynamic animations.
  - If you do not specify a tag in the Animation's "Tag" field, Movicon will use the tag associated to the object for managing Animations (see paragraph on "Assigning Tags to Objects").
12. Close the animation settings window by confirming with "OK".
13. Take a "ToggleButton" from the Toolbox and insert it on the screen to be used for starting and stopping the animation.
14. Assign the button with the same tag defined in the animation settings. By doing this, the 3D object's animation activation tags can be toggles using this button in Runtime.
15. At this point run at test to see if this Animation works by starting up the project in Runtime and use the button to toggle the tags defined. The color of the component selected should animate according to the tag value.

### 6.8.3. 3D Symbol Properties

In addition to the Zoom and View properties, the 3D objects can also be set with properties to rotate and move on the X, Y and Z axis as described below.



## General

**Flow Direction:** This is used to set the flow direction of text and other elements within the object.

**Tooltip:** displays the text indicated by the mouse pointer when used by the user on the control.

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

**Max 3D Zoom:** This is used to set the maximum 3D symbol zooming level.

**Min 3D Zoom:** This is used to set the minimum 3D symbol zooming level.

**Max 3D Rotation Angle X:** This is used to set the maximum clockwise rotation angle during runtime. The operator will not be able to manually rotate the object with an angle major to the one set here in respect to the X axis. The allowed values are in decimals from 0 to 1. Zero value = no constraints.

**Max 3D Rotation Angle Y:** This is used to set the maximum clockwise rotation angle during runtime. The operator will not be able to manually rotate the object with an angle major to the one set in this property in respect to the Y axis. The allowed values are 0 and 1 in decimals. Zero value = no constraints.

**Max 3D Rotation Angle Z:** This is used to set the maximum clockwise rotation angle during runtime. The operator will not be able to manually rotate the object with an angle major to the one set in this property in respect to the Z axis. The allowed values are 0 and 1 in decimals. Zero value = no constraints.

**Min 3D Rotation Angle Z:** This is used to set the minimum anti-clockwise rotation angle during runtime. The operator will not be able to manually rotate the object with an angle less than the one set

in this property in respect to the X axis. The allowed values are 0 and 1 in decimals. Zero value = no constraints.

**Min 3D Rotation Angle Y:** This is used to set the minimum anti-clockwise rotation angle during runtime. The operator will not be able to manually rotate the object with an angle less than the one set in this property in respect to the Y axis. The allowed values are 0 and 1 in decimals. Zero value = no constraints.

**Min 3D Rotation Angle Z:** This is used to set the minimum anti-clockwise rotation angle during runtime. The operator will not be able to manually rotate the object with an angle less than the one set in this property in respect to the Z axis. The allowed values are 0 and 1 in decimals. Zero value = no constraints.

**Max 3D Translate Offset X:** Maximum 3D object movement translation on the X axis.

**Min 3D Translate Offset X:** Minimum 3D object movement translation on the X axis.

**Max 3D Translate Offset Y:** Maximum 3D object movement translation on the Y axis.

**Min 3D Translate Offset Y:** Minimum 3D object movement translation on the X axis.

**3DZoom Tag:** a variable can be associated to pilot the 3D zoom on the symbol. The value is inversely proportional to the applied zoom. Nearer the value to zero, the more the symbol will increase in size. Greater the value, the smaller the symbol will become (A 0 the symbol will result invisible). If you used the "-" sign, the symbol will rotate 180 degrees without influencing the applied zoom level.

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

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

**Menu with Left button:** enables the object's menu to display by using the mouse left button.

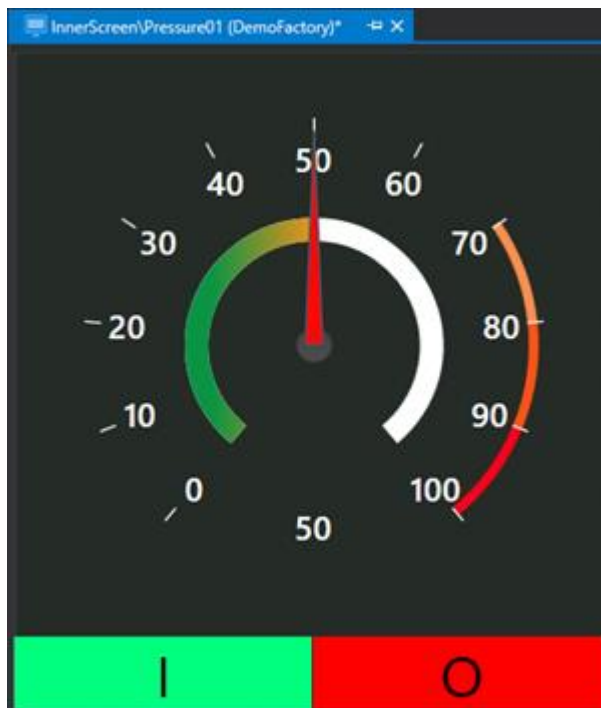
#### 6.8.4. Inner Screens for 3D Animation Graphics

### Inner Screen Animation

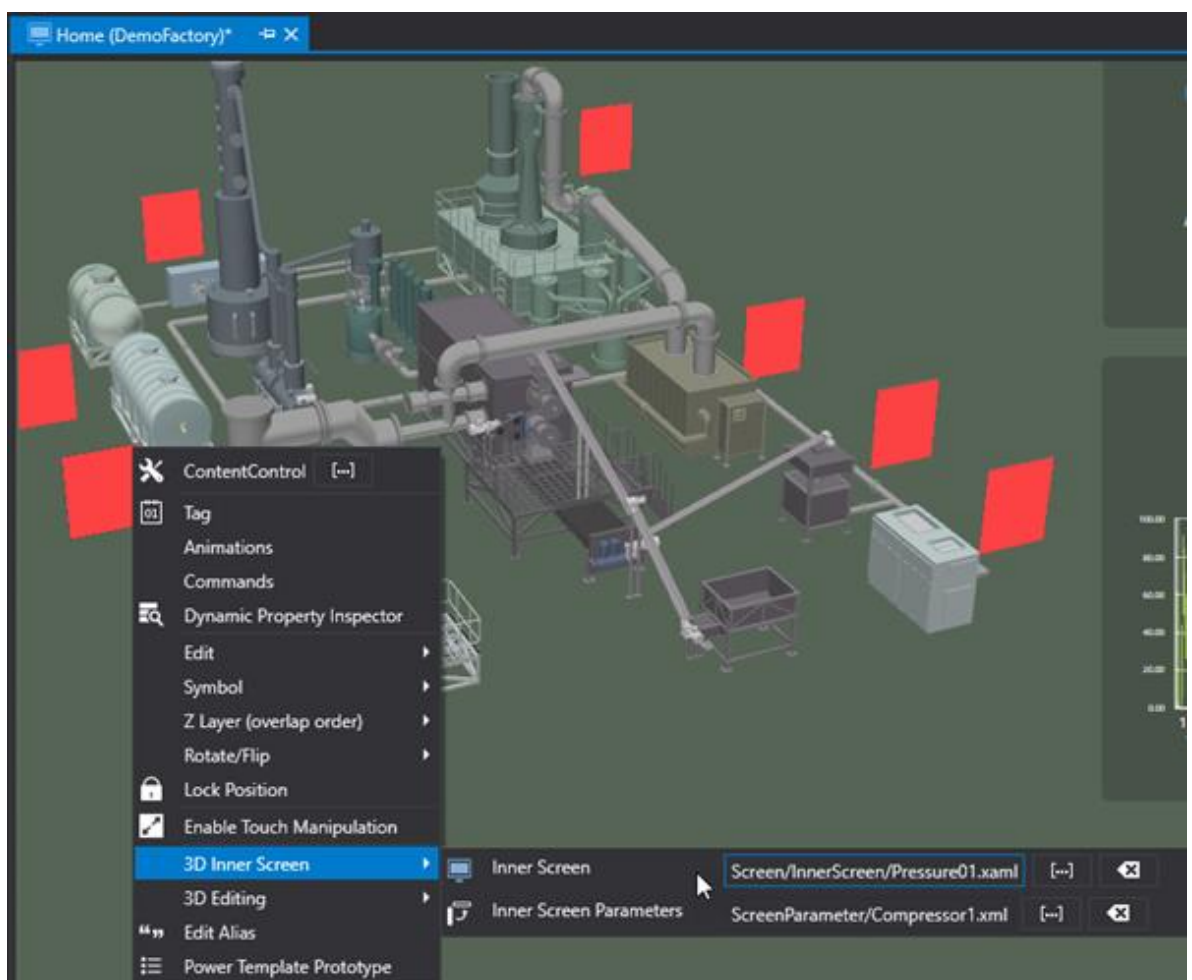
Movicon provides a very interesting feature called Inner Screen to be used in 3D objects. This feature enables you to manage a whole screen within the 3D model, which will be placed on the 'face' of the 3D model's components inserted on screen.

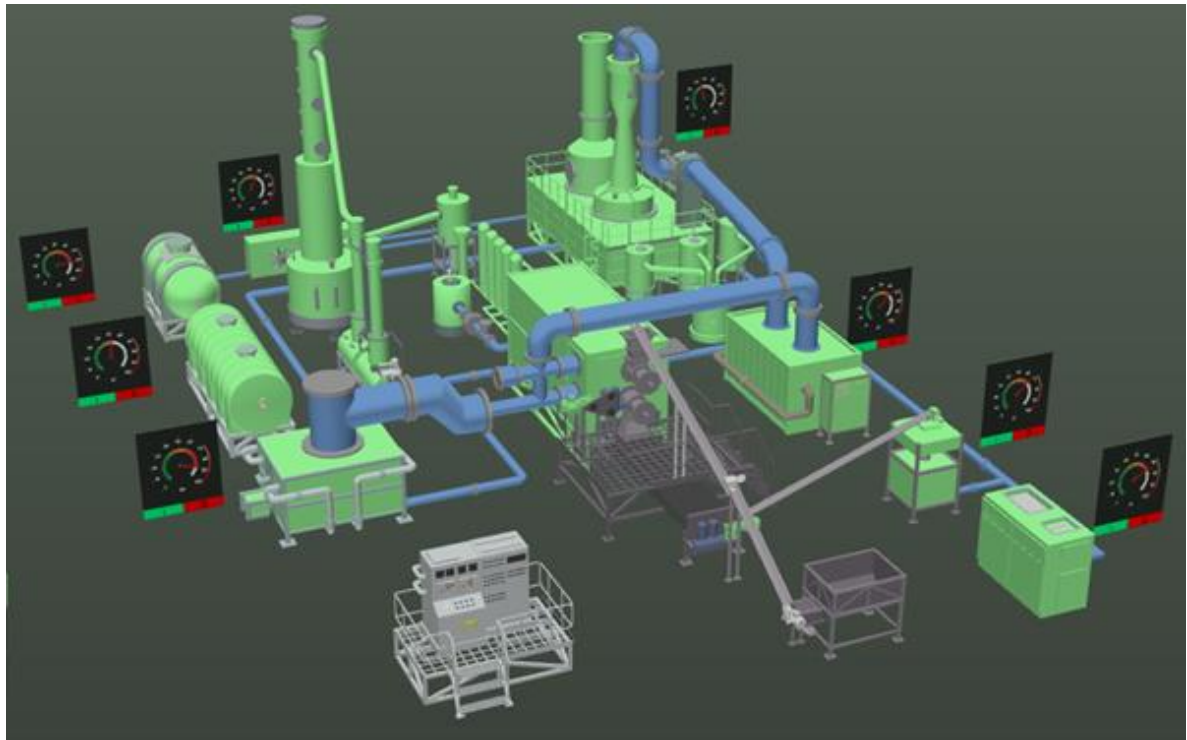
In this way the associated screen will be displayed within the 3D model and will be dynamic in both component content representations and command object interactivity.

- Example: predispose a 600 x 600 pixel sized screen containing a control keypad, display objects, gauges and start/stop buttons all associated with the relevant variables. After having accomplished this you can go ahead and insert this small screen within the 3D model using the "Inner Screen" tab from the object's Common Property Window after having selected the 3D component in which the screen is to be inserted.  
When in project runtime it will be possible to rotate the 3D model and see the 600x600 screen in its position and fully interactive within the 3D model.



*This example shows a screen with display and command objects that will be managed as an 'Inner Screen' within a 3D model scene*

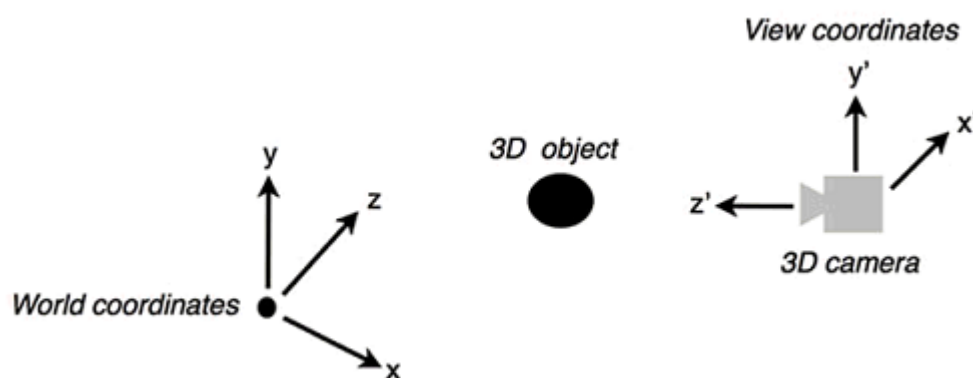




*This example shows how a screen can become an Inner Screen 3D model component. The screen functions as a 3D Inner Screen positioned and fully interactive within the model.*

### 6.8.5. Using the Camera in 3D Graphics

The 3D objects inserted on screen support the 'Zoom' and the 'Views' management according to a freely configurable angle rotation defined "Camera".



### Zooming 3D objects

The displaying of each 3D object inserted on screen can be managed with the 'Zoom In' and 'Zoom Out' commands which refer to the object's scene and not the screen.

In runtime the zoom is applied by pointing and right clicking and keeping the key pressed down on the object to drag the cursor to zoom in and out. When using touch screens, this procedure is done by using the equivalent "pinch & zoom" command.

To perform this operation in edit mode you must first enable the 'Editing' function using the object's command menu and then enable the "**Enable 3D Editing**" and "**Enable 3D Camera View**" items.

### 3D object rotation

The displaying of each 3D object inserted on screen can be managed using the Rotation commands which refer to the scene of each individual object and not the screen containing them.

In runtime the Rotation command is used by pointing and clicking the object and keeping the mouse key down while dragging the cursor.

To perform this same operation in edit mode, first enable the 'Editing' functions using the object's command menu and then the "**Enable 3D Editing**" the "**Enable 3D Camera View**" items.

### Camera settings

More than often it is very handy to record one or more 3D object views so that the runtime user can have different views of the camera's preset angle of vision at their disposition. View recording is performed in project edit mode where the design engineer can position the object's angle of vision as pleased and then used the object's command menu to select "**Edit 3D Camera Position**" item.

This command displays a window through which the user can specify the name to be assigned to the view and thus create a list of recorded display 'positions' that can be selected in runtime.

### Tips & Tricks

#### How to activate a Camera in runtime

During editing mode the custom camera positions can be recorded using the "3D Camera Position" command from the object's command menu. As a consequence, these recorded camera positions can be activated by the user in runtime by right clicking on the object to display a menu that contains a list of the saved camera positions.

#### How to move the Camera of a 3D object in runtime

Click and drag the object with the left mouse key.

#### How to Zoom a 3D object in runtime

Click and drag the object with the right mouse key.

## 6.9. MultiTouch

### 6.9.1. Using multitouch

The modern user interface systems can be equipped with touch screens that support multitouch, as used smartphone or tablet systems, to offer new ways of using industrial HMI systems that are more intuitive and enjoyable to use.

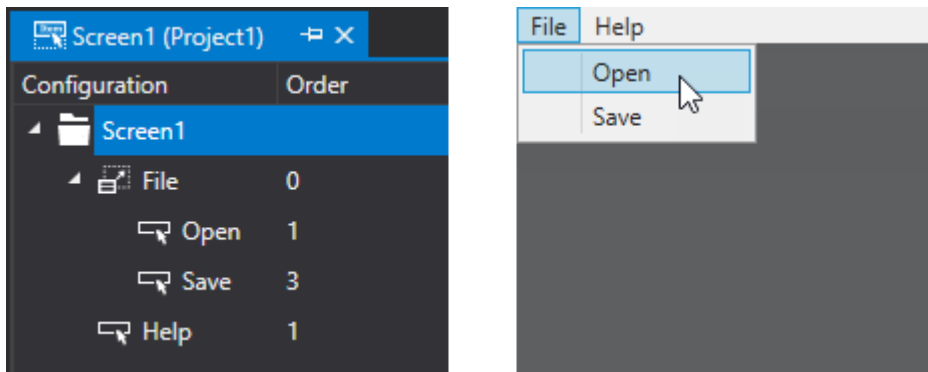
The Movicon.NExT user interface is predisposed natively for Multitouch use.

This Tutorial quickly shows you which are the concepts and the possibilities that can be used to design engineering Movicon screens.



## Manipulating objects

Multitouch use can also be enabled individually for any one of the Movicon graphical objects. By doing this, the user can only manipulate those objects predisposed with the multitouch feature by the programmer for repositioning, enlarging, rotating them on screen as pleased during runtime. To enable objects with "multitouch" manipulation simply select the desired object, activate the command menu using the icon located on its border, and select the **"Enable Object Manipulation"** from the command menu as shown in the image below:



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

**During runtime,** "Manipulation" enabled objects will be predisposed for manipulation within the user interface by using the following procedures:

Click the object desired to display the **command buttons on the left border**. Press the **"Man."** button to enable the object's manipulation. The object will obtain a shadowed background to indicated that is manipulable. The "Man." button is bistable meaning that you must press it again to disable the object's manipulation.

When the object is enabled with manipulation, it can be dragged, positioned, rotated or enlarged as pleased using the above described multitouch techniques but with individual objects.



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



To **restore position**, as defined in design mode, 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:

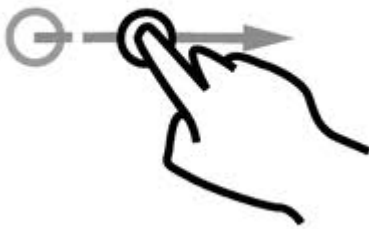


*The command in the Movicon system command bar, shown in this example, enables you to restore manipulated objects back to their original positions.*

### 6.9.2. Multi Touch Gestuality

This guide will briefly go through the concepts and possibilities used in design engineering Movicon screens.

#### Swipe



Swiping is one of the most intuitive and natural user interface experiences in HMI systems where page navigation is much smoother compared to traditional systems.

All you need to do is simply place your fingertip lightly on screen and move it gently to the left or right in order to scroll to the next or previous pages. In order to return to the main page simply tap at the bottom to display the hidden toolbar if not provided otherwise. The main page supports navigation using the touchscreen or scrolling techniques as predefined in the project's general property settings (described in the topic on "Screen Navigation").

#### Pinch & Zoom, Pan



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

The Zoom effect usually enables you to scroll within the panned area or visualize design objects and elements that cannot be seen in normal displays considering that all objects have display properties which function according to the zoom value given to the screen.

## Simultaneous Multitouch Support



A very useful feature to have in the industrial ambient is the possibility to execute commands that have been conditioned to respond to the simultaneous touch of two graphical command objects. For example, to safeguard operators it might be required that an operating command, used for putting dangerous machinery into motion, be invoked by obliging the operator to use both hands to press two buttons, placed at opposite sides of the screen, simultaneously.



The function to simultaneously press two objects is only applied if the command is impulsive type on variable. For all the other command types, only the command of the object pressed first will be executed. In addition, the multitouch function only works for the first two simultaneous touches. This function does not support three or more touches simultaneously but only two at the same time.

## 6.10. GeoScada

### 6.10.1. Geo Page

Movicon.NExT supports a Geo Page feature which integrates geographic maps and geo-localization managements in screens. This feature is used for creating geo-localized projects to enable the handling of maps within the screens which can be navigated and integrated with dynamic data from the project.

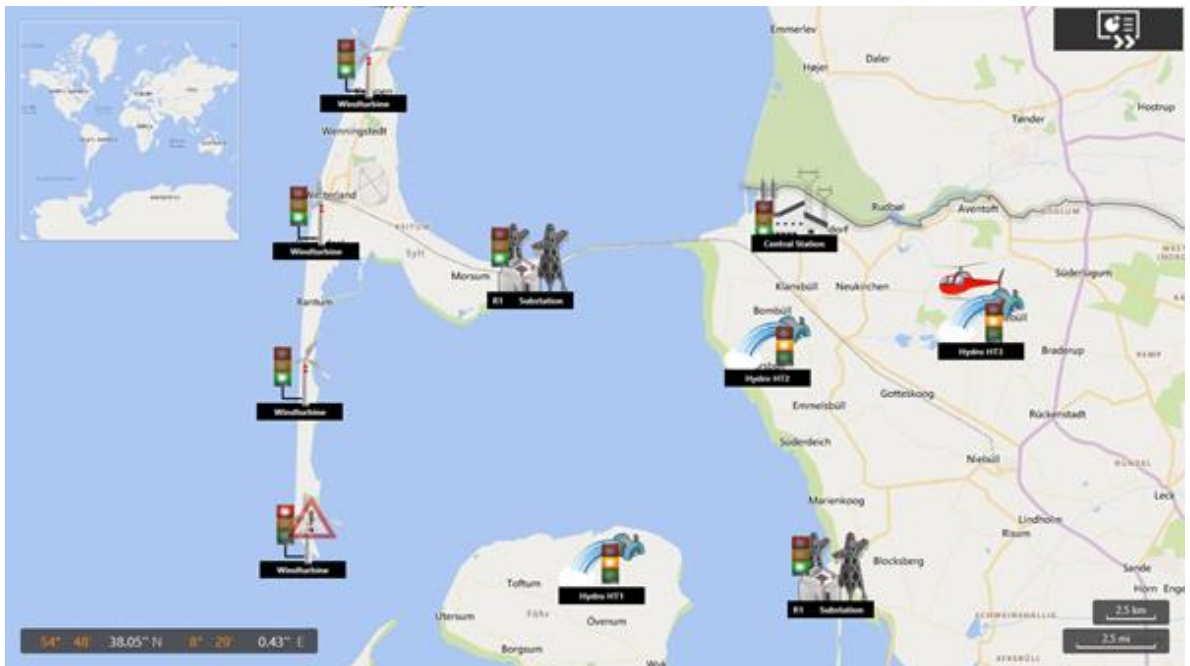


Using maps in a project requires a connection to the internet services. Movicon uses the **Microsoft Bing® Maps** services for default that are subject to the license terms of use as stipulated by Microsoft at

link: <http://www.microsoft.com/maps/product/terms.html>

However the cartographics and map service provider can be changed to OpenStreetMap.org for example or other.

For further information please see the relating documentation.

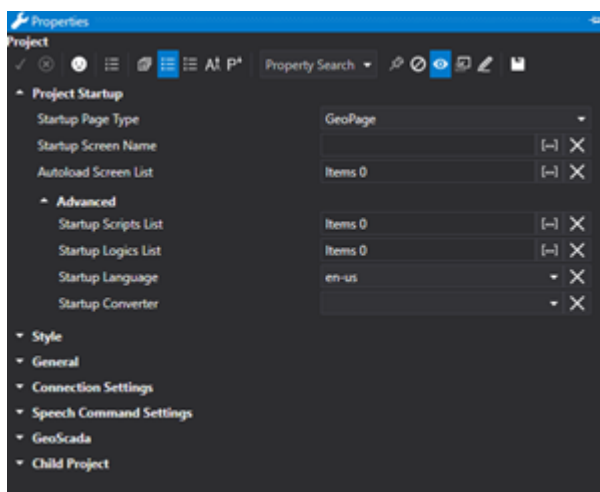


*This is an example of a geo-localization page with dynamic values.*

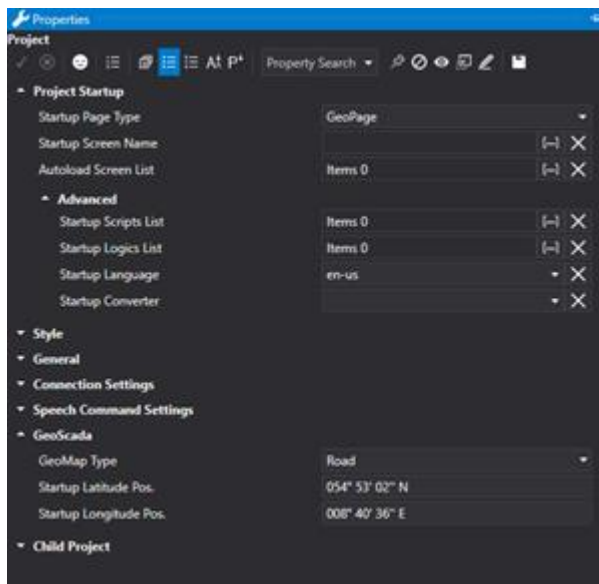
### Configuring GeoPages using the Project's properties

The project's properties can be used to define:

1. By selecting the Geopage option in the 'Startup Page Type' property, the project will startup directly showing the map, as shown in the above image, instead of a Screen. If you choose not to use the GeoPage as the startup page, you can open it by simply associating an "Open Map" command to a button or an event to do so.

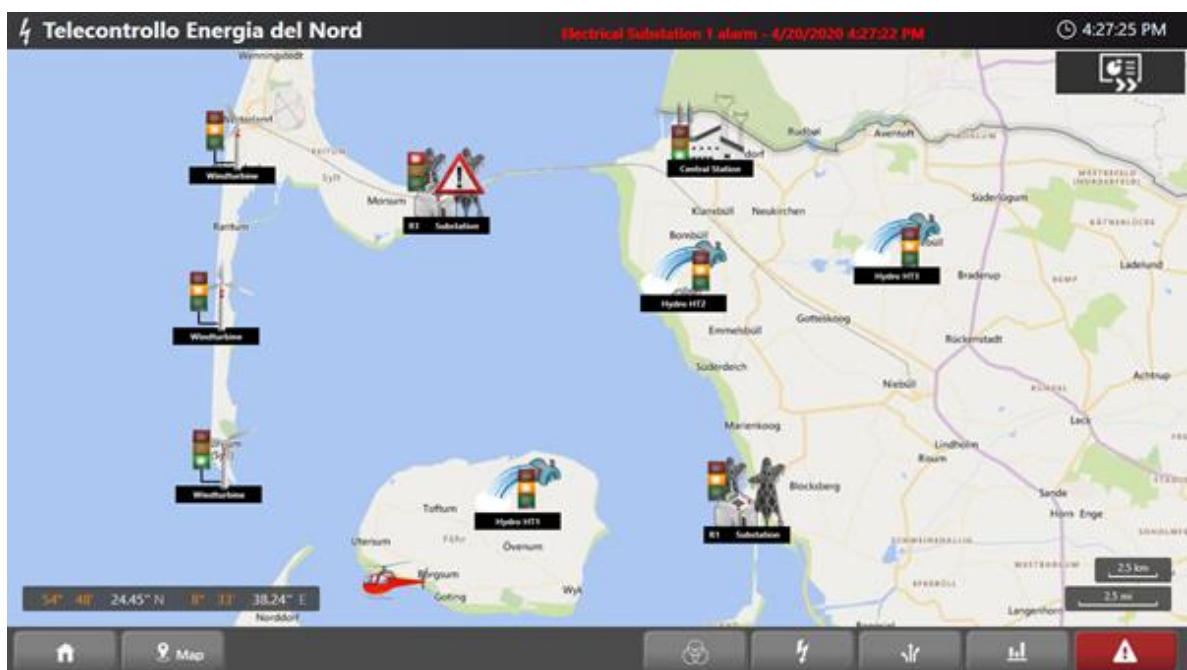


2. The "Latitude" properties that identify in which predefined position to open the map.
3. The "GeoMap Type" property is used to pick which map to use. The choices are: Road Map, Geographic and hybrid.



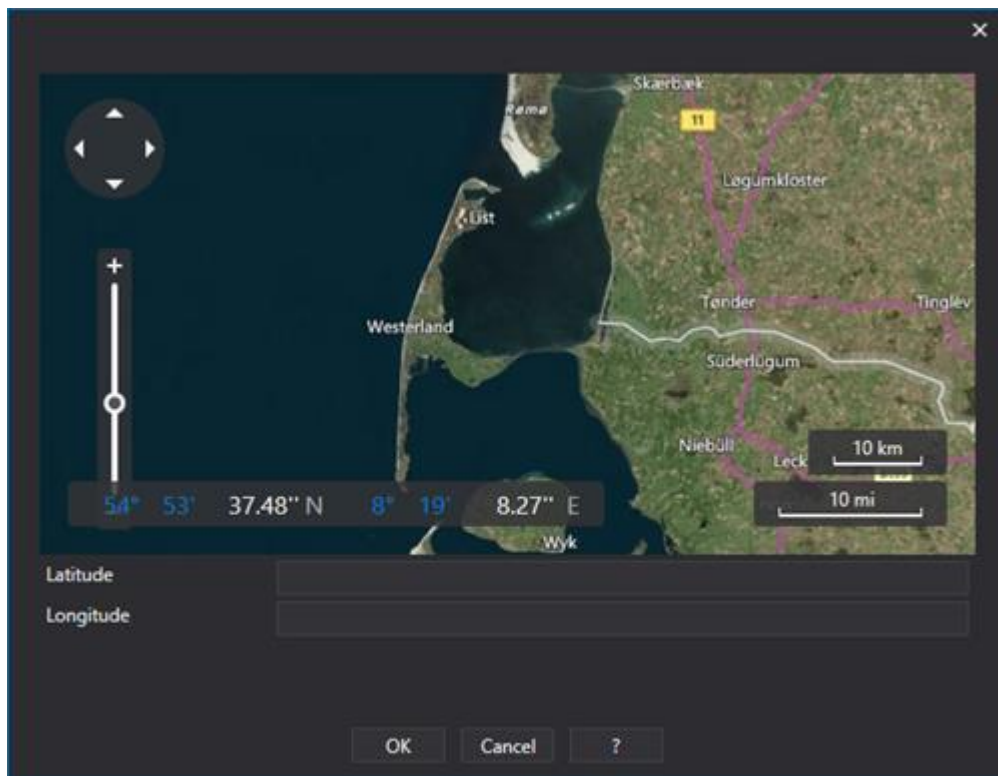
## Configuring GeoPages with Layout Screens

Layout screens can be used to display geographic maps in the project whether displayed at startup or displayed by using a command. In this case, the map will be considered like any other Screen displayed in the centre and other layout screens can be used for other functions, like the one shown below, by coordinating and integrating them in the Project appropriately.



## Latitude and Longitude Selection Tool

Movicon provides a selection tool called "Geo Localization Editor" to facilitate the identification of the Latitude and Longitude properties. This tool can be access from the Options > Edit Geo Location or from the optional "Project" toolbar which displays the window below:



This window is used to navigate around the map and when clicking on any point, the tool will give the Latitude and Longitude coordinates. It is also possible to enter a specific address by typing it in the "Address" edit box and then confirm with the "Search" key. The map will automatically go to the position of the geographic address and return the Latitude and Longitude values. The selected coordinate values can then be used in the project's properties.

### 6.10.2. Screens in Geographic Maps

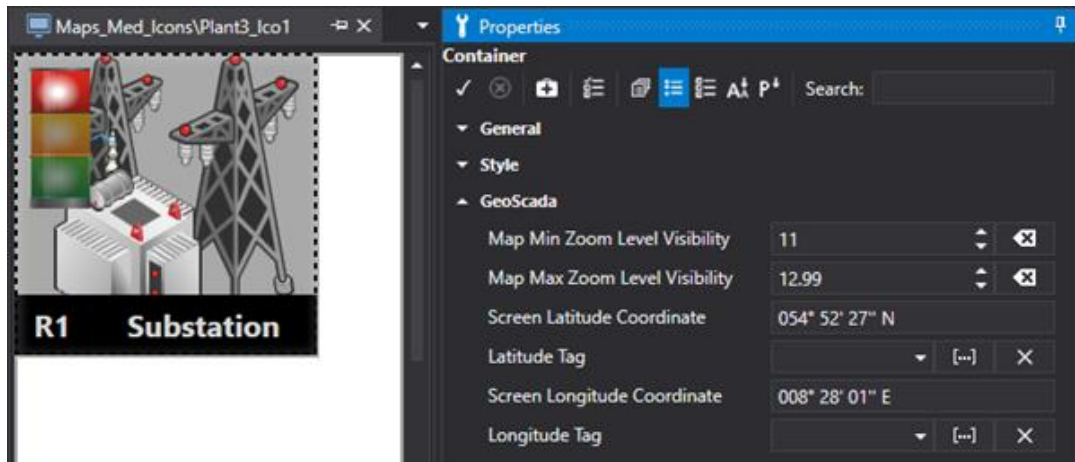
The Geo Page screen containing the geographic map can be integrated with additional screens and relating content.

Contents of other project screens can be represented dynamically on the map according to the geographic coordinate points defined in the screen's properties. The screen shot below shows a map using this criteria:



The designer can therefore create a specific screen for each point on the map with the desired size in pixels. The screen will be displayed on the map showing its graphical contents including command interactivity.

The properties to use for displaying the screen on the map are:



#### Map Min Zoom Level Visibility

Indicated the map's minimum zoom value (between 1 and 20) within which the screen is displayed.

#### Map Max Zoom Level Visibility

indicates the map's maximum zoom value (between 1 and 20) within which the screen is displayed.

#### Screen Latitude Coordinate

Specifies the "Latitude" to position the screen on the map with.

#### Screen Longitude Coordinate

Specifies the "Longitude" to position the screen on the map with.

#### Latitude Tag

This is used to make the screen's latitude dynamic on the map by connecting it with a Tag.

#### Longitude Tag

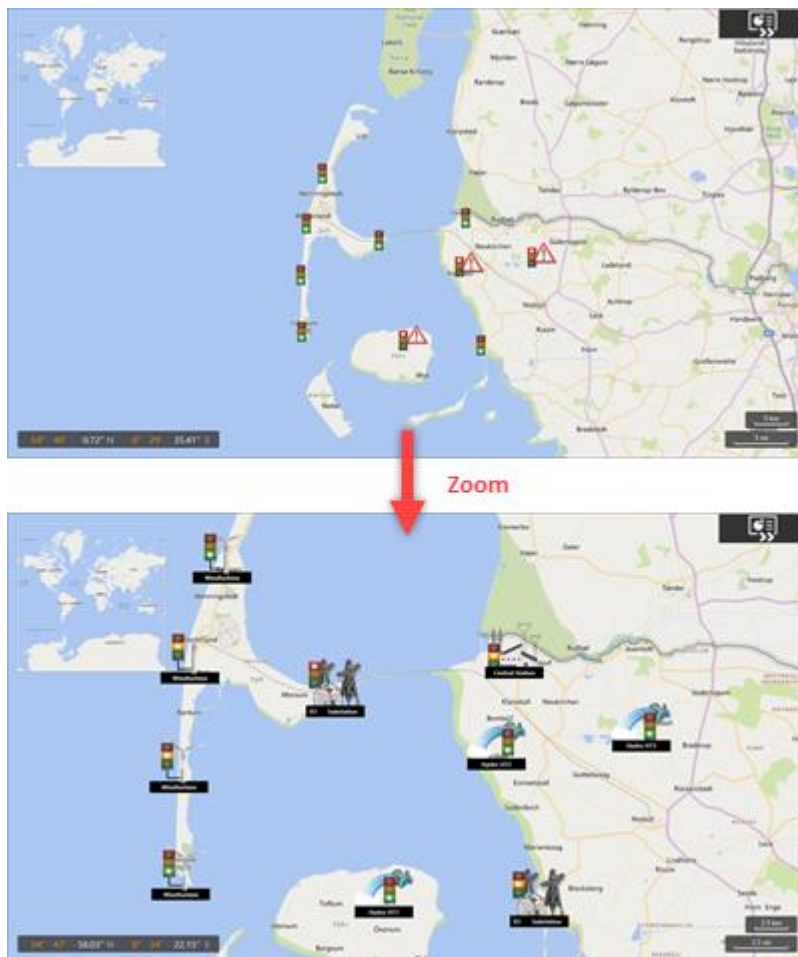
This is used to make the screen's longitude dynamic on the map by connecting it with a Tag.

The properties above provide the availability of two functionalities as described below.

#### Visualizing objects with the zoom:

The "Map Min Zoom Level Visibility" and "Map Max Zoom Level Visibility" properties make it possible to zoom screens to make them visible at the desired level according to the amount of information you wish to be displayed. This is particularly handy when needing only to display little information by using a minimum zoom level (zoom level = 1) or needing to display screens containing a lot more detail using a higher zoom level (zoom level = 20).

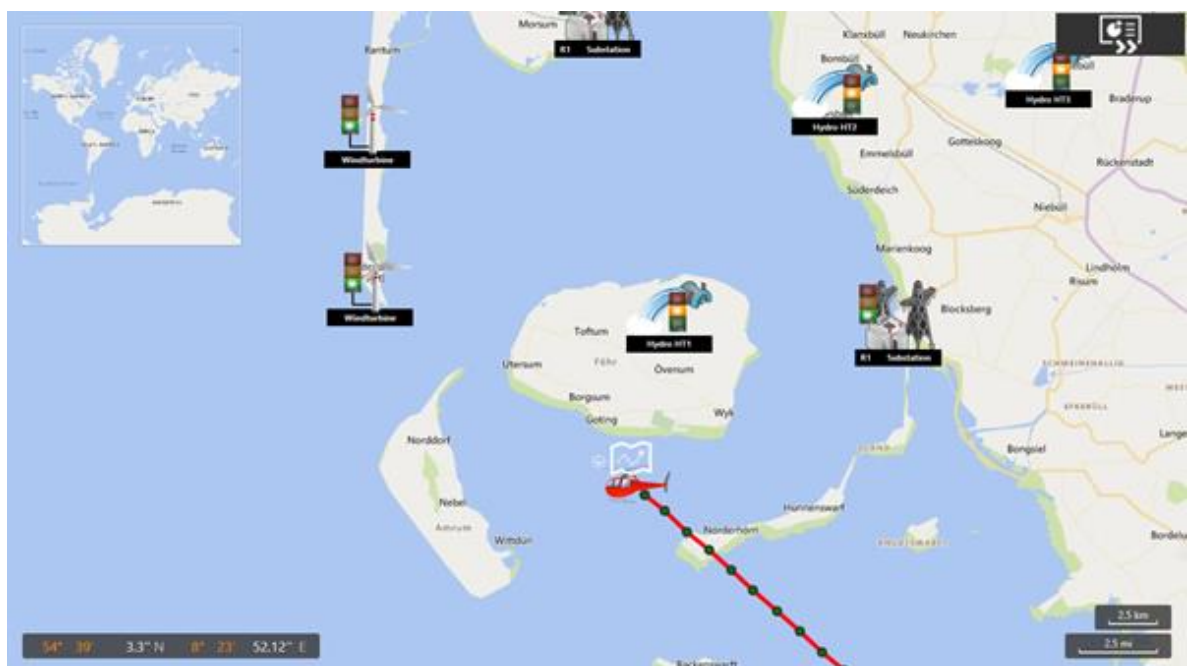




### Visualizing moving objects

The "Latitude Tag" and "Longitude Tag" properties are handy when needing to display moving objects on the map. This makes it possible to keep track of the object's movement by simply logging the two tags associated to these properties in a historian.

In this case, the screen on the map will show a menu through which it is possible to display the track of the route taken by the moving object over time.



### 6.10.3. Alternative Providers and Local Maps

Movicon.NExT uses the Microsoft Bing® Maps services but also allows you to change the default map provider or save maps in your own system in order to load them from a local path. This is very handy in those cases where the only internet connection you have is not always available.

#### Using the Open Street Map

In the "DocumentManagers" folder within the Movicon.NExT installation path, you will find the "ScreenManager.dll.config" file through which it is possible to change the map provider from Microsoft Bing Map to Open Street Map. This is done by simply editing the file to change the "UseOpenStreetMapProvider" option from False to True as indicated in the image below:

```
<setting name="UseOpenStreetMapProvider" serializeAs="String">
  <value>True</value>
</setting>
```

#### Using maps locally

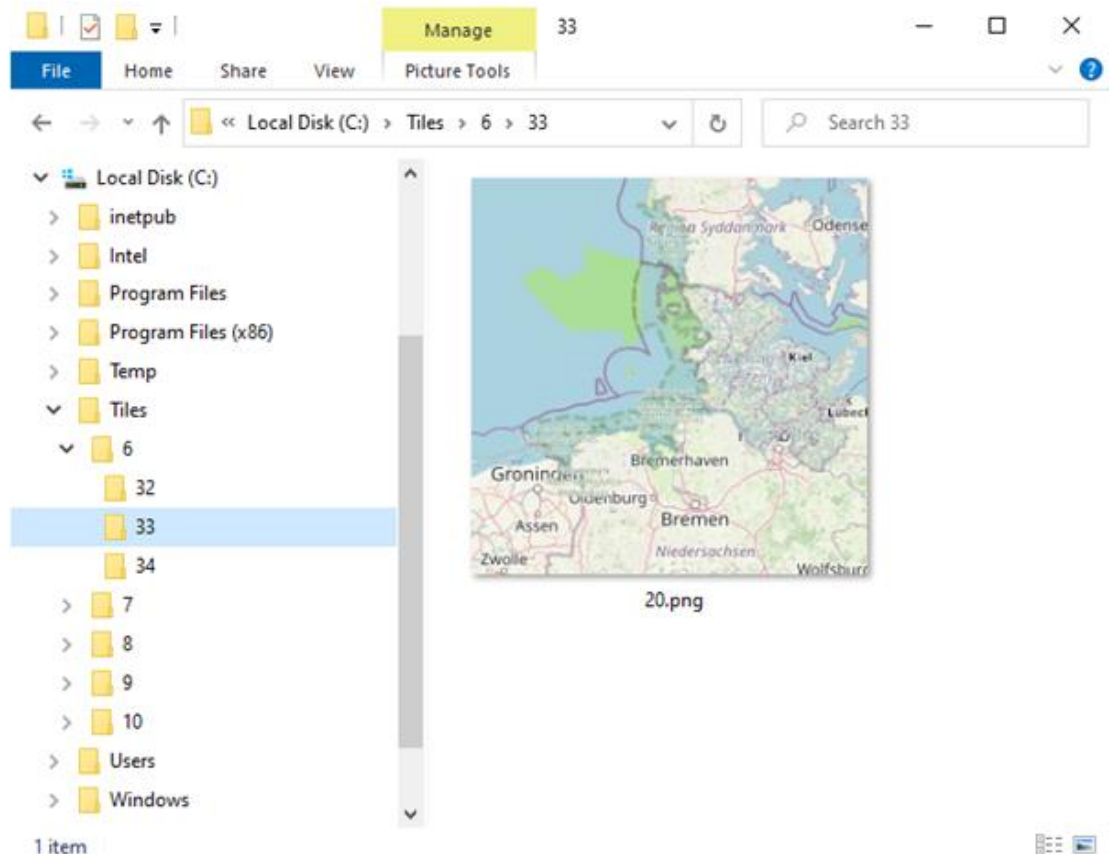
In order to use the maps locally to avoid needing to use an active internet connection simply:

1. Download the map of the interested area in ".osm.pbf" format.  
Files which are continuously updated are available, for example at, <http://download.geofabrik.de>, which allows you to select the various areas in the world. Obviously, the larger the area, the more time it will take to download the data and, as a consequence, to process them (for example, a file containing the map of Italy is more than 1 GB and takes several hours to elaborate).
2. Generating the Tiles of downloaded maps  
To generate the Tiles which compose the map, you will need to use Maperitive for example, which is a free program that you can get from <http://maperitive.net/>.  
Once started up, Maperitive allows you to open the map in ".pbf" with the File menu >Open map sources item and generate the map Tiles using the Tool menu > Generate Tiles.  
The Tiles are generated in the "Tiles" folder within the Maperitive application path with a 6 to 10 zoom level. If you need to generate Tile with a wider zoom range (maximum zoom range is 1 - 20) you will need to execute the "generate-tiles [parameters]" command from the command line.



Attention: each zoom level quadruples the number of tiles generated, therefore generating tiles with a wide ranging zoom level may take a long time and the tiles may occupy several Gigabytes of space on the disk.





### 3. Configuring Movicon.NExT to use the generated Tiles

In the "DocumentManagers" folder within the Movicon.NExT installation path you will find the "ScreenManager.dll.config" file where you need to do is modify the items:

- 3a. "UseOpenStreetMapProvider" setting it to "True" as shown below:

```
<setting name="UseOpenStreetMapProvider" serializeAs="String">
  <value>True</value>
</setting>
```

- 3b. "CustomMapDataProviderUrl" setting it as "file:///C:\\Tiles\\{tileLevel}\\{tileX}\\{tileY}.png" as shown below:

```
<setting name="CustomMapDataProviderUrl" serializeAs="String">
  <value>file:///C:\\Tiles\\{tileLevel}\\{tileX}\\{tileY}.png</value>
</setting>
```

The path defined in the "CustomMapDataProviderUrl" indicates to Movicon.NExT to search for the map filed installed in the "C:\\Tiles\\" subfolders, where:

{tileLevel} indicates the folder containing the Tiles with a specific zoom level

- {tileX} indicates the folder containing the X row of the quadrant.
- {tileY} indicates the name of the png file of the Tile at column Y of the quadrant.

#### 6.10.4. Maps in WebHMI

The WebHMI applications support GeoScada technology to allow browsers to display screens based on geographical maps with overlaying dynamic project information according to the features described in the chapter on GeoScada.

The Maps' functionality is always configured in the WebHMI Server device by considering that you may also need to configure the system accordingly.

Maps managed in WebHMI up to version 4.1 are in vectorial format and should be downloaded, prepared, and installed on the server device as described below. This solution does not require an internet connection connected to a GeoMap services provider. Instead, local server systems can be used without connecting to GeoMap services providers.

However, an option to use internet connections to GeoMap services providers has been provided as from version 4.2. as it does with the Movicon.NExT Desktop version.

This solution offers more detailed maps which do not require any downloading or installation, but only an internet connection to the provider with an adequate bandwidth..

### Configurazione Mappe Vettoriali per WebHMI

To customize vectors maps in WebHMI you will need to:

1. Download an existing map of the areas available from <https://download.bbbike.org/osm/> . If you cannot find a specific area you can try this other website <https://extract.bbbike.org/> to extract the area desired. You must download the selected area in OSM format.
2. The OSM file must be converted to the geojson format. This can be done in your PC or on-line. Please refer to the GDAL program manual for converting files off-line: <https://gdal.org/download.html#current-releases>. . Below is an example of some commands used for converting the file:

- `ogrinfo <downloaded_file_name>.osm`  
This command returns the available layers within the file, such as:

*Had to open data source read-only.*

*INFO: Open of `osm'*

*using driver `OSM' successful.*

*1: points (Point)*

*2: lines (Line String)*

*3: multilinestrings (Multi Line String)*

*4: multipolygons (Multi Polygon)*

*5: other\_relations (Geometry Collection)*

- `ogr2ogr -f GeoJSON map_points.json <downloaded_file_name>.osm points`
- `ogr2ogr -f GeoJSON map_lines.json <downloaded_file_name>.osm lines`
- `ogr2ogr -f GeoJSON map_multilinestrings.json <downloaded_file_name>.osm multilinestrings`
- `ogr2ogr -f GeoJSON map_multipolygons.json <downloaded_file_name>.osm multipolygons`
- `ogr2ogr -f GeoJSON map_other_relations.json <downloaded_file_name>.osm other_relations`

As an alternative, you can use an on line conversion service by first exporting the maps for example from "Openstreetmap" as "\*.osm" files.

3. To edit the downloaded maps, install the QGIS software (<https://www.qgis.org/en/site/forusers/download.html>) to display and edit/filter the layers previously created in order to add a new layer (such as a grey background for example). When a

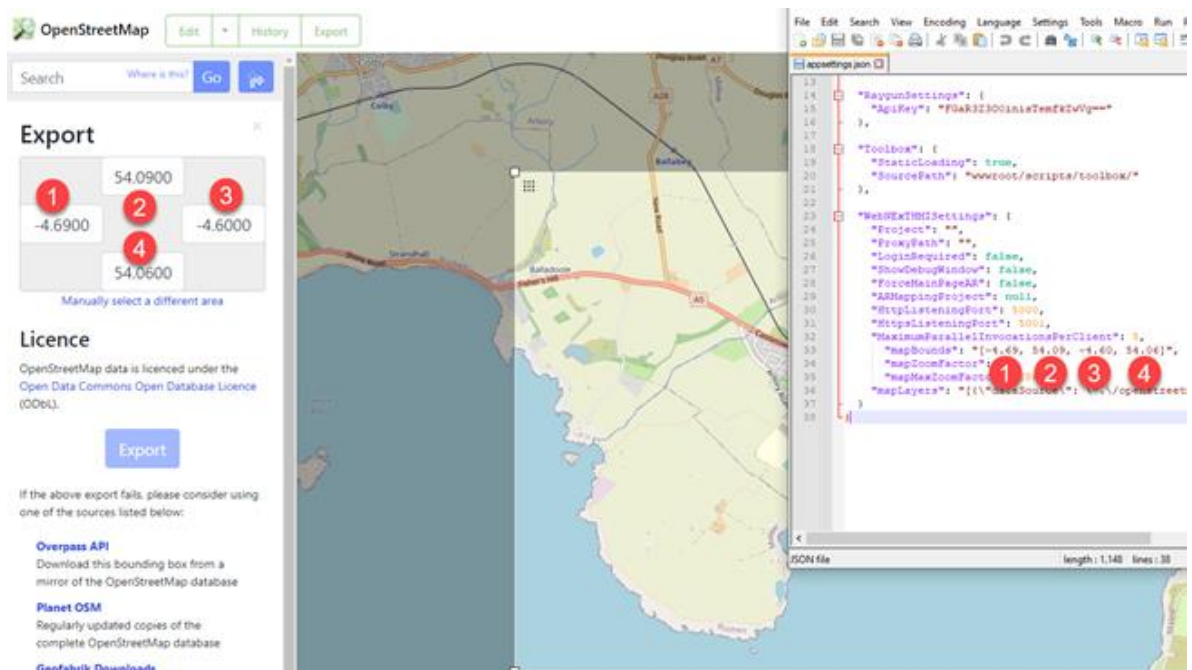
filter is applied to a layer (such as "buildings" = 'yes'), this layer can then be saved with a new name.

4. Copy the.json files to the webhmiserver's wwwroot (C:\Program Files\Progea\Movicon.NEXT x.x\WebHMIServer\wwwroot), and possibly in a subfolder like in our example: .../openstreetmap/
5. Edit the appsettings.json file within the C:\Program Files\Progea\Movicon.NEXT 4.1\WebHMIServer path to use it and specify other parameters such as:

```
"WebNExTHMISettings": {
  "Project": "",
  "ProxyPath": "",
  "LoginRequired": false,
  "ShowDebugWindow": false,
  "ForceMainPageAR": false,
  "ARMMappingProject": null,
  "HttpListeningPort": 5000,
  "HttpsListeningPort": 5001,
  "MaximumParallelInvocationsPerClient": 5,
  "mapBounds": "[-4.69, 54.09, -4.60, 54.06]",
  "mapZoomFactor": 1,
  "mapMaxZoomFactor": 256,
  "mapLayers": "[{\"dataSource\": \"\\.\openstreetmap/landcover.json\",
  \"hoverEnabled\": false}, {\"dataSource\": \"\\.\openstreetmap/buildings.json\",
  \"hoverEnabled\": false}, {\"dataSource\": \"\\.\openstreetmap/roads.json\",
  \"hoverEnabled\": false, \"label\": {\"enabled\": true, \"datafield\": \"name\"}}]",
}
```

These parameters are used to do the following:

- **mapLayers:** Allows you to declare a string array of the filename of the layer to be used.
- **mapBounds:** Allows you to define the geographical area displayed at startup.



*Clicking on the export button will give you the possibility to display/edit the coordinates of the points that define the area to be exported. These points can be used within the configuration file to define the displayed area.*

- **mapZoomFactor**: Specifies a number that is used as the initial map zoom factor.
- **mapMaxZoomFactor**: Specifies the maximum map zoom factor.

When finished, set the project's "Startup Page Type" to Geo Page, and deploy the project using the project's CreateWebHMI (SVG/HTML5) command.

## **Map Raster with connection to Services Provider configuration**

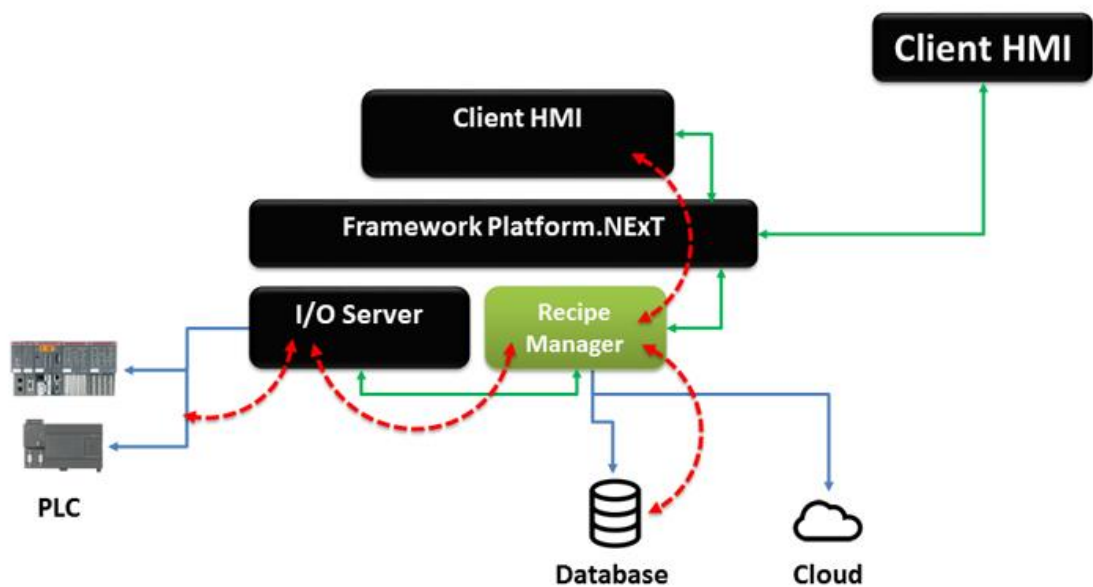
Work in progress, available as from version 4.2

# 7. Recipe Manager

## 7.1. Introduction to using Recipes

Often manufacturing plants need to record parameters and set points of different product types on file and store in an archive system so that they can then be modified when and as needed to activate production runs of different products. This is what is commonly known as production recipe management.

The Movicon.NExT Recipe management allows you to define system or process data groups, values and set points that must be recorded and managed in database archives. They can then be selected by name (recipe index or title) when needed and activated by transferring them to the PLC or field devices.



- This diagram shows a function block scheme of the Movicon.NExT Recipe Manager. This module manages connections towards the database and display objects along with direct connectivity with field devices.

### Recipe Connectivity to PLC and Database

For instance, production processes whose end product is the result of a mixture composed of different components, each with a specific measure, necessitate a solution where such components and measures can be established in production recipes. Once the recipe has been established with the right ingredients needed to obtain the desired end product, it can then be activated and processed accordingly.

This concept can be applied whenever it is necessary to enter, save, print and activate data whenever required.



Movicon.NExT completely manages the use of the Recipes in automatic using a functional model which comes included with the system and is called the 'Recipe Manager'. This module can be configured by means of using the 'Recipe' resource for each recipe in the project tree structure.

A Movicon.NExT Recipe is provided with a database connection to manage historical log filed where the recipe data values are recorded.

All the recipe's data values are also connected directly to the PLC so that data transferral can be managed autonomously. This direct connectivity method reduces the time to transfer data because there is no need to pass through any I/O Data Server polling management to synchronize data in order to complete the transfer.

## Database Connectivity

The recipe is provided with the option to connect to a database file through the properties which the recipe index can be set with. The recipe will use the Microsoft SQL Server as set for default in the project's general settings if not specified otherwise.

However the recipe's properties can also be used to customize the connection to another database different to the one used for default if desired.



A recipe is saved in the database in one or more tables associated to each other. Usually these tables consist of a main table and a number of secondary tables, one for each data group. Each recipe instance has a data record in each table.

In the figure below there are two tables containing data from the Recipe1 recipe composed of 3 data fields (Data1, Data2, Data3) and a group composed of another 2 data fields (Data1Group1, Data2Group2): in this figure we can see that 3 different recipe instances have been recorded and have been called formula1, formula2 and formula3 respectively.

Risultati

Messaggi

	Recipe1_ID	Recipe1	Recipe1_CreationDateTime	Recipe1_ActivationDateTime	Data1	Data2	Data3
1	30D7517D-5782-433C-A11D-19F535EB7074	formula 3	2016-04-01 14:04:41.2500000	2016-04-01 14:07:34.6600000	3	3	3
2	1CF69842-5D01-4A5C-9A81-2C91A33597F2	formula 2	2016-04-01 14:04:21.8430000	2016-04-01 14:07:48.1030000	2	2	2
3	92119677-C9ED-48EE-B027-5AD96161085C	formula 1	2016-04-01 14:04:06.8200000	2016-04-01 14:07:56.4100000	1	1	1

Risultati

Messaggi

	Recipe1_ID	Data1Group1	Data2Group1
1	30D7517D-5782-433C-A11D-19F535EB7074	3	3
2	1CF69842-5D01-4A5C-9A81-2C91A33597F2	2	2
3	92119677-C9ED-48EE-B027-5AD96161085C	1	1

## PLC Connectivity

The recipe can be set with a physical I/O address of a PLC or field device through its index and element properties (group, data value). This will ensure that data is transferred from the database to the PLC and vice-versa in one unique read or write operation towards at least one communication driver set in the project.

This type of connection management ensures the **safe transfer of one unique data block**, that is synchronized and autonomous. This is because It does not pass through the I/O Data Server and therefore does not carry the risk of being fragmented when polled by priority.

Substantially this maximizes security of read and write operations towards the devices.

The "Start Address" can be set at individual "DataValue" level and "Group" level as well as in the actual recipe. The hierarchical order with which the connection address towards the device is used is as follows:

1. Address defined at "datavalue" level
2. Address defined at "group" level
3. Address defined at "recipe" level

Please also consider that use of the address at Group or Recipe levels is only possible in certain conditions and according to the Communication Driver being used. For example, if using a Communication Driver where the address is set in the variables' properties, each data value must also must be set with the same address. Conversely, if the driver can only be used with absolute addresses and there are several data exchange areas being used (eg. Modbus which have Coil, Holding Register areas, etc.), then setting the address at Group level requires that the group data be homogenous and with consecutive addresses.



It is the programmer's job to use Movicon.NExT correctly to set addresses at each field, data group and entire recipe level by taking into consideration the communication protocol specifications used.

## 7.2. Recipe Management

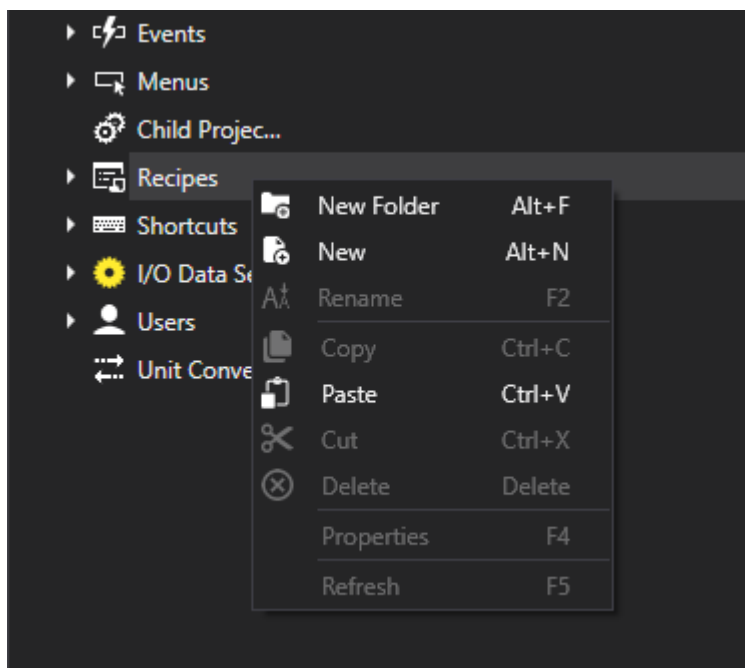
A Movicon.NExT Recipe is composed of one or several data values that can be divided into different Groups while maintaining reference to the Recipe Index.

The Recipe data values are saved in the database and connect to the field devices by means of the I/O Data Server's Communication Drivers. This enables recipe values to be loaded from the database and transferred to the PLC or to connected field devices or vice-versa whereby values can be read from the device and saved in the database with the set Recipe name (index).

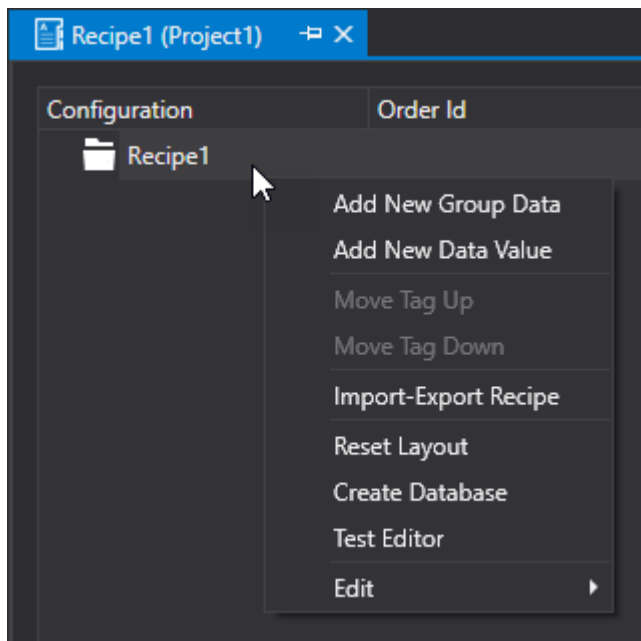
The recipe structure is represented by a tree structure within its own Editor as shown above. The recipe is comprise of various as described below:

### Adding a new Recipe

To create a new Recipe you can either use the Project menu > Recipes > Add New or the Recipe resource's contextual menu from the Project tree structure as shown in the screenshot below:



Once the recipe has been created, by using the Project menu > Recipes or the contextual menu within the resource, you can add Groups, Recipe Fields and perform other operations as shown in the screenshot and explained below:



## Recipe Management Commands

### Add New Group Data

This command is used to add a new group in the recipe. Adding a Group in the recipe structure will allow you to create the basis for grouping data which is useful for dividing the many recipe fields in an orderly manner and for defining a specific physical starting address for each individual group.

### Add New Data Value

This field allows is used to add a new Data Value in the recipe. The fields in the Recipe structure correspond to data values, such as set points, that constitute the actual recipe's values. Therefore, each field defines a data value which will be memorized in the recipe based on its index.

### Move Tag Up

This command is used to move a variable up one place in the recipe tree structure.

### Move Tag Down

This command is used to move a variable down one place in the recipe tree structure.

### Import-Export Recipe

This command is used to export the recipe's configuration to a csv file to continue editing with external tools (e.g. Microsoft Excel) and subsequently to import it from the .csv file. For further details, please refer to the topic on "Recipe Import & Export".

### Reset Layout

This command is used to resets the recipe's layout.

### Create Archive

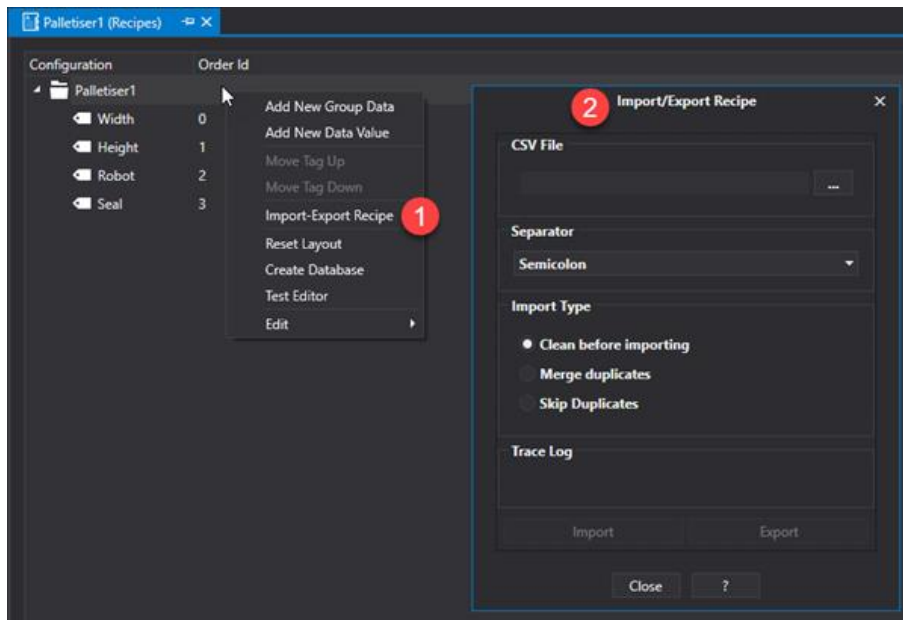
This command creates the recipe's data structure in the associated database by cancelling the existing one, if any, along with its data contents.

### Test Layout



This command displays the system window with the recipe's graphical interface as it will appear to the user in runtime using the system Pop Up window.

## 7.3. Recipe Import & Export



This tool is used to import and export Movicon.NExT recipe structures in CSV. file format.

This file can also be edited using an external editor and then be reimported within the workspace afterwards.

### Properties

**CSV File:** Indicates the name and path of the CSV file which will be used during the import/export procedures.

**Separator:** This is used to select the column separator used in the CSV. file. The options are:

- Colon
- Comma
- Semicolon
- Tab
- Custom (this lets you specify a custom character to use as the separator)

**Import Type:** The import modes are:

- **Clean before import:** Eliminates all existing contents within Movicon.NExT before importing the new contents.
- **Merge Duplicates:** Merges duplicates that are detected while being imported.
- **Skip Duplicates:** This allows you to specify which duplicated fields to be skipped when found in the CSV file.

**Trace Log:** logs all what happens during file import/export procedure.

## 7.4. Recipe Properties

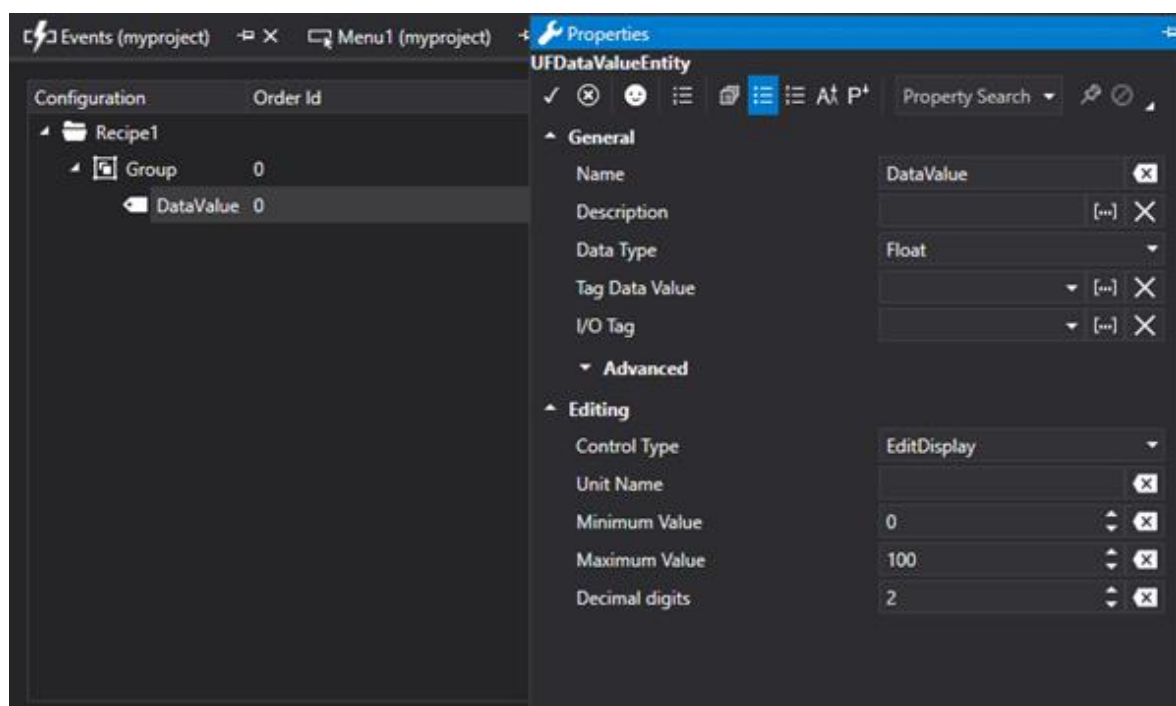
The following properties can be edited for each defined recipe:

- Data Value
- Groups
- General Properties

by simply selecting each component of the recipe to use its properties window.

### Data Value Properties

The Data Values (Fields) of a Recipe represent the values that are exchanged with the device. Their configuration is similar to that of a normal Tag. The properties are those shown below:



#### Name

Name of the Recipe Field.

#### Recipe Table Name

This property allows the user to customize the name of the recipe table.

#### Description

This parameter is used to enter a description of the Recipe Field. The description text can either be entered as liked and it will be displayed in the Recipe Viewer object.

#### Data Type

This parameter is used to define the data of the Recipe Field.

#### Maximum Length

This is only available when "String" has been selected as "Data Type" and allows you to define the maximum string length with characters.

## Tag Data Value



Do NOT use the Tag containing the physical I/O address for exchanging recipe data with the device as the Tag to associate to this field.

In this field you can select a I/O Data Server Tag to use for all the operations to manage recipes at Runtime.

This field is used in combination with developing a custom Recipe management interface (as an alternative to using the "Recipe Viewer" object from the ToolBox). Please refer to the topic on "Using Recipes at Runtime".

## I/O Tag

This field is used to select a Tag from the I/O Data Server that will be used to exchange data with the devices.

The Tag you select must have a Physical I/O address to read/write the value on the device.



Using this modality, the recipe's Data Values will be exchanged with the device using the driver in aggregated mode.

However, this does not guarantee that the device data exchanged with device comes in one unique block.

When you need to exchange data with the device in one unique block, you will need to specify a Physical I/O Address manually in the Recipe's or Group's General Settings.

## Advanced

### Initial Value

This is used to enter a default value that will be assigned to the Recipe Field when a new Recipe is created. If no default value is assigned, the numeric fields will be initialized with zero and the String field will be nulled.

### Allow Null

Enabling this will allow the Null value to be assigned to the Recipe field.

### Enable I/O on Device

When enabled, this allows the Recipe Field value to be exchanged with the device by using a Physical I/O Address or using a communication driver without going through the I/O Data Server.

When enabling this parameter, it will be possible to define a specific physical I/O address for the field manually.

### Encoding

This is only available when the Data Type is String type and allows you to select the encoding type to using for saving on database.

### Physical I/O Address

This is only available when the "Enable I/O on Device" is active, and it allows you to specify the address of the data exchange area with the device by directly using a communication driver without passing through the I/O Data Server.

The settings of the address are the same available for the project's Tags. As an alternative, you can also define a "Physical I/O Address" in the General Settings of the Recipe or Group and the ""I/O Tag" of the recipe's Data Values.



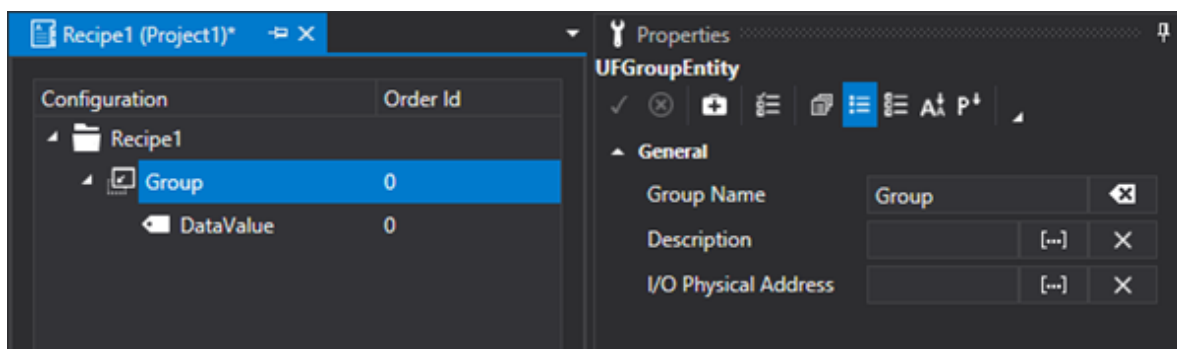
Using this modality, the recipe's Data Values will be exchanged individually with the device, therefore, the recipe data will NOT be exchanged in one block. When you need to exchange data with the device in one unique block, you will need to specify a Physical I/O Address manually in the Recipe's or Group's General Settings.

## Editing

The properties belonging to the "Editing" group allow you to define the User Control type to use within the "Recipe Viewer" object.

## Group Properties

The Group properties of a Recipe, in addition to grouping data in an orderly fashion, allows you to specify the starting address to be referred to for exchanging data of fields that belong to the Group. In this way, a diverse number of data exchange areas can be assigned to each Group.



### Group Name

Indicates the Group's Name.

### Description

This is used to enter text to describe the Group as pleased and which will be displayed in the Recipe Viewer object.

### Physical I/O Address

This is only available when the "Enable I/O on Device" is active, and it allows you to specify the address of the data exchange area with the device by directly using a communication driver without passing through the I/O Data Server.

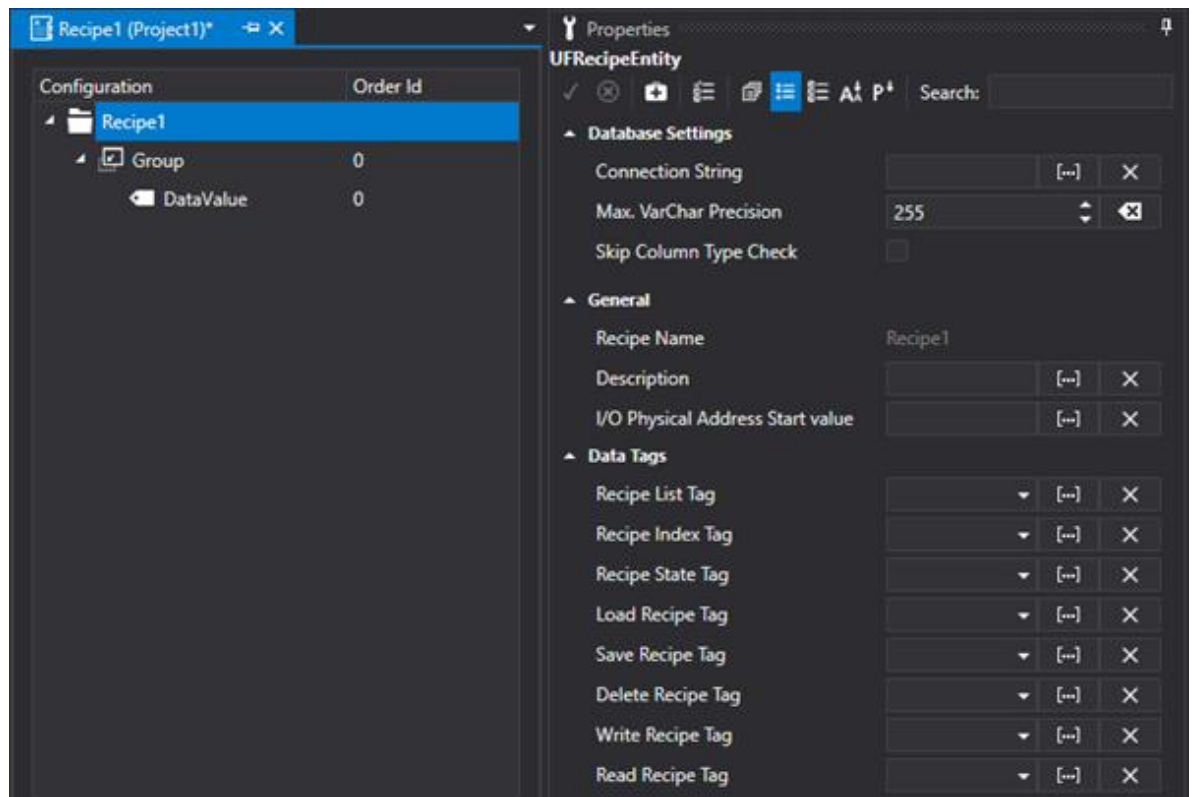
The settings of the address are the same available for the project's Tags. As an alternative, you can also define a "Physical I/O Address" in the General Settings of the Recipe or Group and the ""I/O Tag" of the recipe's Data Values.



The use of this modality allows the group's data to be exchanged with the device in one unique block but this method is only available for those drivers that use a NON symbolic address. When defining the start address only, the offset address of each Data Value will be automatically calculated according to the order of the Recipe Fields that, therefore, must be in the same order of the device's memory areas.

## General Recipe Properties

The General Recipe Properties, which are essential for the normal recipe use, mainly allow you to set database connection parameters, define the recipe's physical I/O starting address and temporary Tags.



### Recipe Name

Indicates the name of the Recipe. This field is read only. If you wish to change the name of the Recipe, you will need to use the Project Explorer Window.

### Description

Indicates the Recipe Description.

### Physical I/O Start Address

This is used to specify the start address of the data exchange area with the device by directly using a communication driver without passing through the I/O Data Server. The settings of the address are the same available for the project's Tags. As an alternative, you can also define a "Physical I/O Address" in the General Settings of the Recipe or Group and the "I/O Tag" of the recipe's Data Values.



The use of this modality allows the recipe's data to be exchanged with the device in one unique block but this method is only available for those drivers that use a NON symbolic address.

When defining the start address only, the offset address of each Data Value will be automatically calculated according to the order of the Recipe Fields that, therefore, must be in the same order of the device's memory areas.

### Recipe List Tag

This field is used to select a string type Tag in which the list of recipes saved on Database will be showed at runtime.

Each recipe name will be separated by the "pipe" (|) character in the string.



The Tag associated to this field together with the Tag associated to the "Recipe Index Tag" field can use a combo box with which recipes saved on the database can be displayed and selected.

### Recipe Index Tag

This field is used to select a string type Tag with which the name of recipe to be loaded from the Database can set at Runtime. The name of the recipe defined in this Tag will also be used by the "Recipe Commands" for reading and writing the values.



The Tag associated to this field together with the Tag associated to the "Recipe List Tag" field can use a combo box with which recipes saved on the database can be displayed and selected.

### Recipe State Tag

This field is used to select a UInt32 Tag type in which the Recipe State will be reported at Runtime as reported below:

- Bit 0: Recipe Manager Initialization
- Bit 1: Stop Recipe Manager
- Bit 2: Loading Values from DB in progress
- Bit 3: Saving Recipes Values on DB in progress
- Bit 4: Deleting Recipe from DB in progress
- Bit 5: Reading Values from Device in progress
- Bit 6: Writing Values in Device in progress
- Bit 7: Importing Recipe Values from File in progress
- Bit 8: Exporting Recipe Values from File in progress
- Bit 9: Not Used
- Bit 10: Not Used
- Bit 11: Not Used
- Bit 12: Error while updating Recipe List Tag
- Bit 13: Error while updating State Tag
- Bit 14: Error while updating Recipe Value Tag
- Bit 15: Error while saving Recipe on DB
- Bit 16: Error while deleting Recipe in DB
- Bit 17: Error while reading Values from Device
- Bit 18: Error while writing Values in Device
- Bit 19: Error while importing Values from File
- Bit 20: Error while exporting Values from File
- Bit 21: Recipe Command Tag Connection Timeout
- Bit 22: Recipe Value Tag Connection Timeout
- Bit 23: Not Used
- Bit 24: Not Used
- Bit 25: Values were saved on DB successfully
- Bit 26: Recipe was deleted from DB successfully
- Bit 27: Values were read from Device successfully
- Bit 28: Value were written on Device successfully
- Bit 29: Recipe values were imported correctly
- Bit 30: Recipe values were exported correctly
- Bit 31: Recipe values were read from DB correctly

### Load Recipe Tag

This field is used to select the Tag that allows recipe values present in the Recipe Index Tag (the active recipe) to be loaded from the database when value assumes a value different from zero. Value will be automatically zeroed by the system after the operation.

### Save Recipe Tag

This field is used to select a Tag that allows the recipe, whose name is present in the Recipe Index Tag, to be saved on DB with the values that are currently present in the variables associated to that recipe. This happens when the Tag assumes a value that is different from zero and which will be zeroed automatically by the system after the operation has completed.

### Delete Recipe Tag

This field is used to select a Tag that allows the Recipe, whose name is present in the Recipe Index Tag, to be deleted from the DB with the values currently present in the variables associated to that recipe. This happens when the Tag assumes a value that is different from zero and which will be zeroed automatically by the system after the operation has completed.

### Write Recipe Tag

This field is used to select a Tag that allows the values of the recipe, whose name is present in the Recipe Index Tag, to be sent to an external connected device (e.g. PLC). This happens when the Tag assumes a value that is different from zero and which will be zeroed automatically by the system after the operation has completed.

### Read Recipe Tag

This field is used to select a Tag that allows values to be read from an external connected device (e.g. PLC) and which will be assigned to the recipe whose name is present in the Recipe Index Tag. This happens when the Tag assumes a value that is different from zero and which will be zeroed automatically by the system after the operation has completed.

### DB Connection String

This field is used to define the connection string to be used for recording Recipe data on database. If you do not specify anything in this field, Movicon will use the "Historian Default Connection" string from the I/O Data Server Settings.

### Max. VarChar Precision

This field is used to specify the maximum number of characters that the recipe string can have. This parameter operates on both on the recipe name and on its defined string values.

### Skip Column type control

Disables the consistency check on the data type between the columns of the recipe table and the type of Tag on the DB during runtime execution.

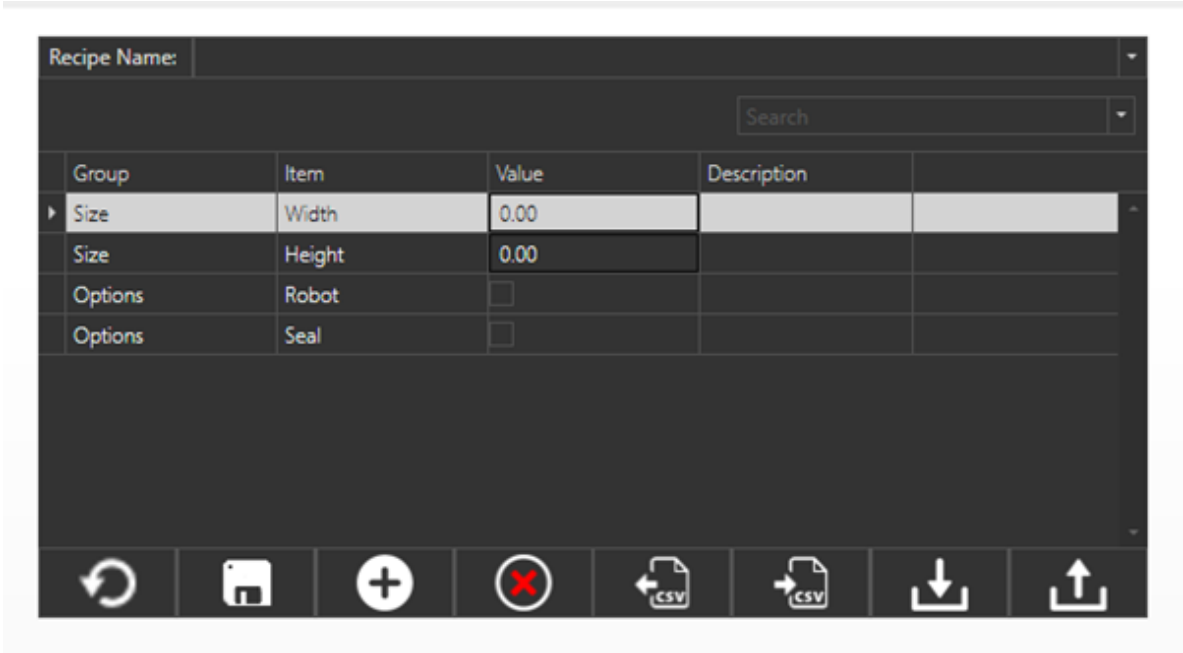
## 7.5. Using Recipes at Runtime

Recipe operativity can be managed in two ways during Project Runtime:

- By using the native "Recipe Viewer" object from the Toolbox
- By using a custom graphical interface made possible by using the Recipe management tags.

# Using the Recipe Viewer object

The Recipe Viewer object can be used to manage the operativity of recipes at Runtime and it reports the Data Values in table format as shown below with a toolbar at the bottom containing the commands to manage the recipe with.



## Recipe Selection Combo-Box

There is a combo-box at the top of the object which allows you to:

- Define a new Recipe index to be memorized.
- Select (and then call) a Recipe Index previously saved on database.

## Data Value Grid

There is a grid in the middle of the object which reports a Recipe field on each row. According to the configuration of each Data Value's "Editing" group properties, it is possible that a Display or Checkbox or other be used in the Value Column for imputation.

## Load Button

The command loads the Recipe Index from the Database displaying its values in the grid.

## Save Button

This command takes the values from the grid and saves them in the Recipe Index selected from the combo-box on database.

## New Button

This command allows a new Recipe Index be created with the values display in the current grid.

## Remove Button

This command allows you to remove the currently selected Recipe Index from in the Database.

## Import Button



This command imports a Recipe Index from a .csv file whose structure must be coherent with that of the recipe's.

#### Export Button

This command exports the Recipe Index selected from the combo-box to a .csv file.

#### Read Button

This command reads the data of the device (PLC, etc.) and used them to update the fields of the active Recipe Index.

It will also be possible to save a new Recipe with a different index or overwrite the active one with the "Save" command.

#### Write Button

This command writes the Index Recipe data to the device (PLC, etc.).



Values written on the PLC are always those which are present on the Database. This means that in order to write modified Recipe values on the PLC, they must first be saved on Database using the Save command.

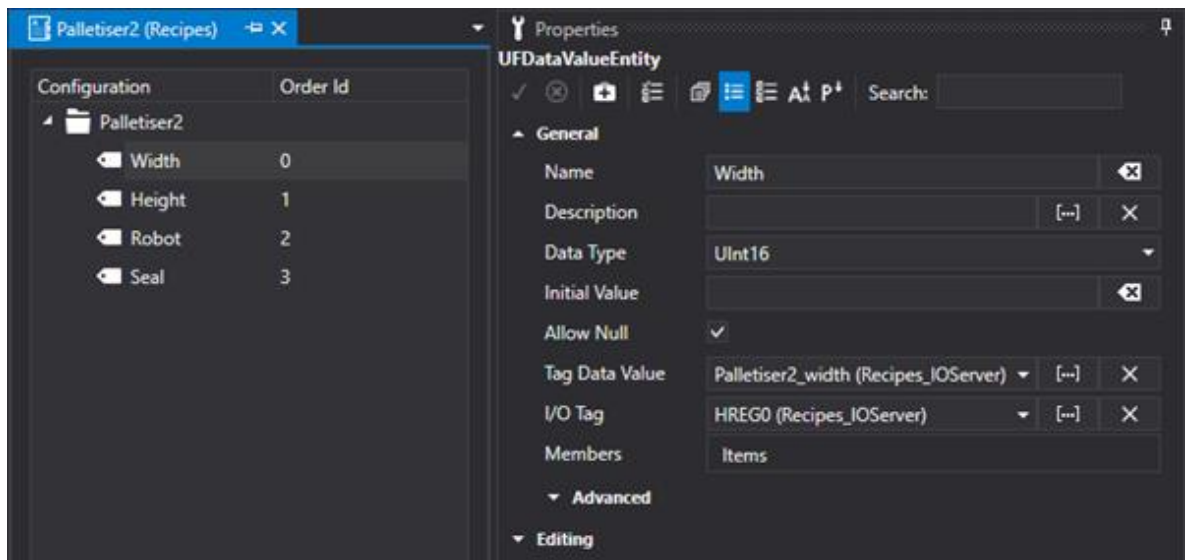
## Using custom graphical interfaces

Recipe operativity can also be managed at Runtime using a custom graphical interface.

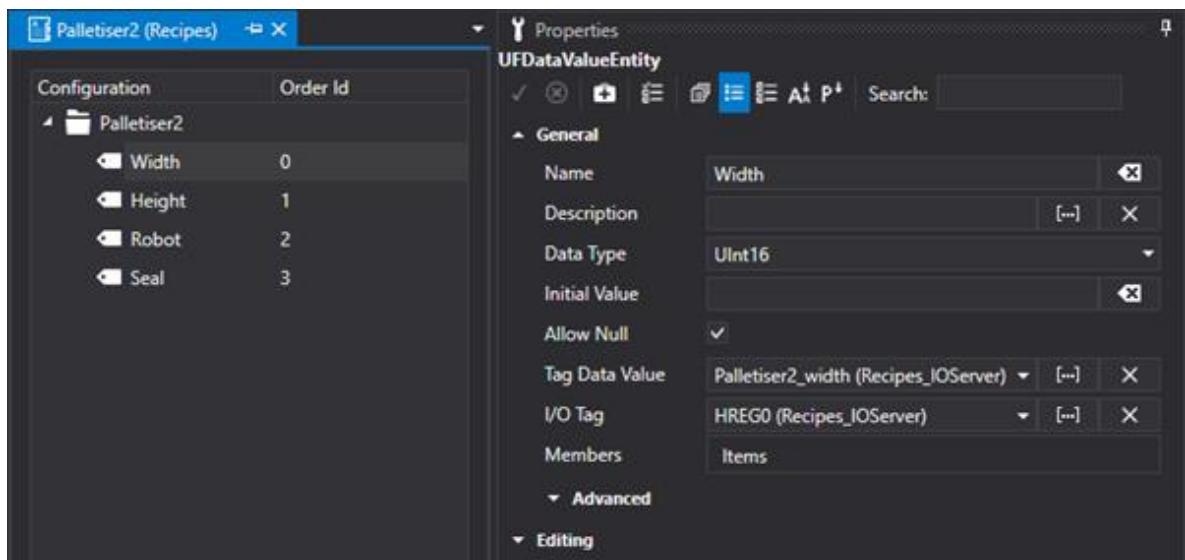
The image shows a custom graphical interface for a device named 'Palletiser2'. The interface is displayed on a blue background. At the top, there is a dark grey rounded rectangle containing the text 'Palletiser2'. Below this, the interface is organized into two columns. The left column contains three controls: 'Set Selection' with a dropdown menu showing 'Set1', 'HREG 0' with a numeric input field showing '8', and 'COIL 0' with a boolean input field showing 'True'. The right column contains three controls: 'Set Name' with a text input field showing 'Set1', 'HREG 1' with a numeric input field showing '10', and 'COIL 1' with a boolean input field showing 'True'. Each input field has small up and down arrow icons on its right side, indicating they are interactive controls.

To create a custom interface you will need to:

1. Use a temporary Tag to link to the "Tag Data Value" property of each one of the Recipe's Data Values.



2. Use the recipe management tags to link to the Recipe's properties.



The use of tags gives you maximum flexibility in managing recipes. Be reminded that:

- The maximum recipe layout graphics customization possible; the tags can actually be represented on a screen using the graphics tools provided in the Movicon.NEXT environment.
- The Possibility to manage recipes in background using, for example, script code, and therefore without the presence of an active user to operate through the Client interface.
- The possibility to create embedded recipes. One recipe is defined embedded when its Index Tag is associated to the field of another recipe. In this way, it will be possible to subdivide an articulated and complex recipe into sub-recipes, all connected to each other.

## 8. Scheduler Manager

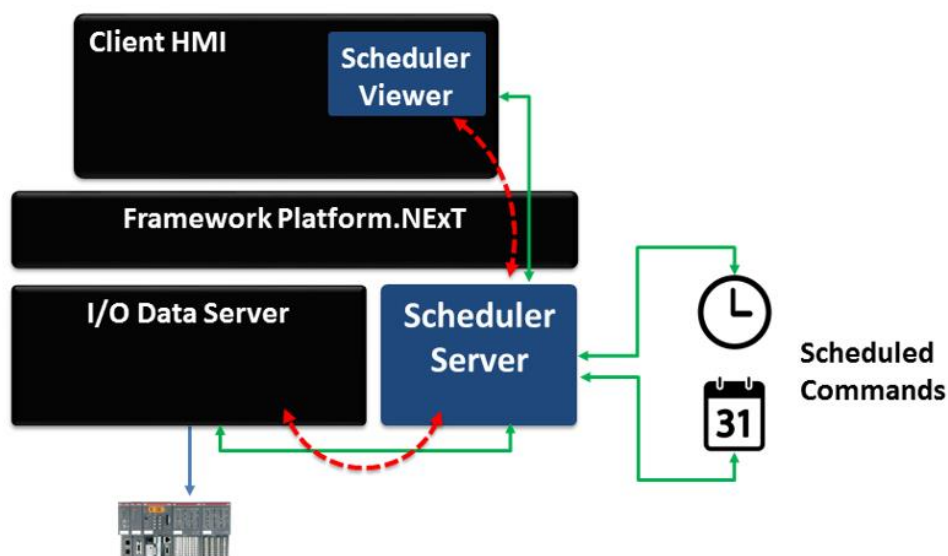
### 8.1. Introduction to using Schedulers

The Movicon.NExT project's 'Scheduler' resource is the tool used for configuring command executions in the project based prefixed times, weekly plans or calendar dates. The Resource is based on a Scheduler Server module therefore an applied module that is part of the platform.

The scheduler is very handy to use when needing to activate commands based on a specific time or date especially when the need to activated in the plant on a regular basis for a certain length of time. It also saves a great deal of time when creating projects with command that needed to be repeatedly activated at preset time ranges in the plant.



The Schedulers use the time set on the PC in which the Scheduler Server is run.

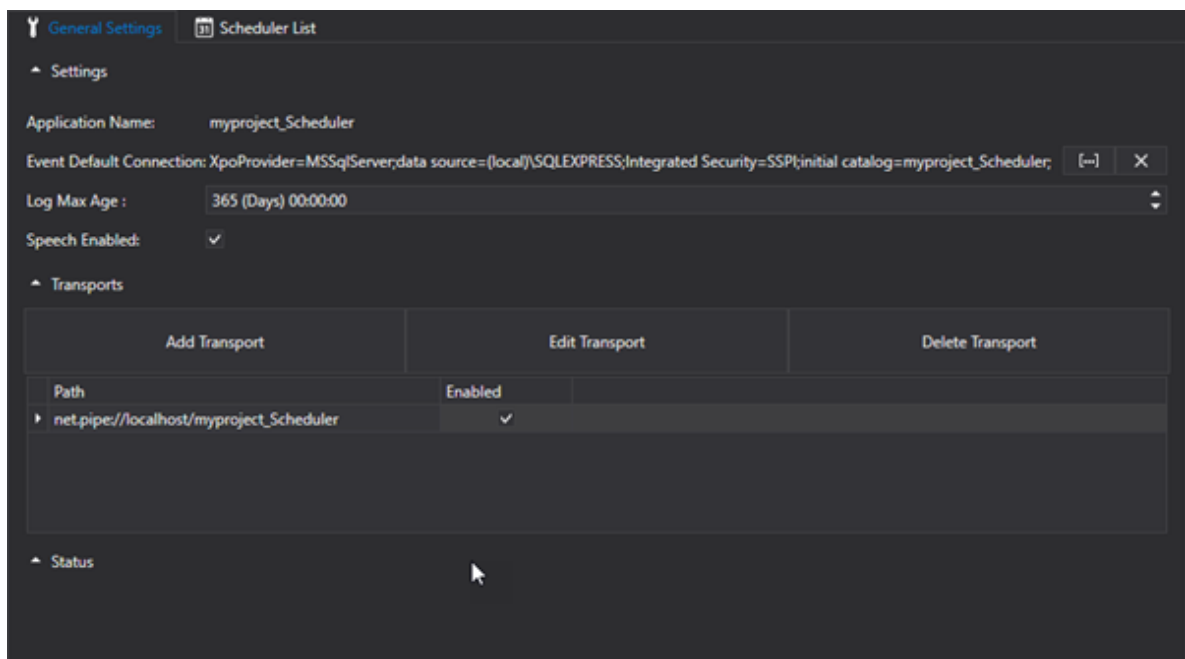


The Scheduler offers great operative advantages in any type of application especially in those sectors (e.g. building Automation, Scheduled Maintenance ecc..) that usually require commands to execute automatically in function with preset time schedules. Just think of all those turning on/off light commands and central heating systems for instance. These operations, even though simple, can be performed using the appropriate resource assigned the task to slim down and speedup the setting procedures of these types of commands.



The Movicon.NExT Scheduler Manager is in fact a OPC UA Server that when started up connects as a Client to the Project's Data Server to access its tags. The graphical Scheduler objects also connect as Client but to the Scheduler Server instead.

The Scheduler Server can be started up manually using the "Start Server" command from the toolbar, or automatically at the Movicon.NExT project startup in which it was configured.

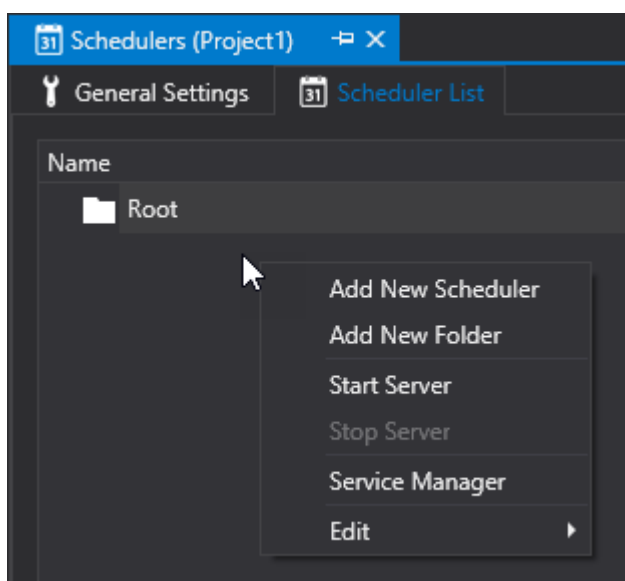


The Scheduler Resource can be opened project edit mode by double clicking on the relative resource in the Movicon.NExT project's tree structure. The resource will open in the workspace in edit mode with two main areas, **General Settings** and **Scheduler List** that can be accessed by selecting the relative Tab on the top border of the resource area.

When selecting the Scheduler List tab you will be able to insert new scheduler objects or display and modify those that already exist by using the Property window.

## Commands for inserting schedule events

The command used for inserting and managing Scheduler events are made available with a right click as shown in the screenshot below.



### Add New Event

The command is used for inserted a new Event after which must be configured accordingly.

### Add New Folder

This command is used for creating a new Folder within which the events are to be inserted. These Folders are only for project organization purposes and can be nested in many levels.

### Start Server

The Scheduler is in fact a OPC UA Server that connects to the Movicon.NExT project's I/O Data Server as Client. The Server can be started up manually by using the Start command.

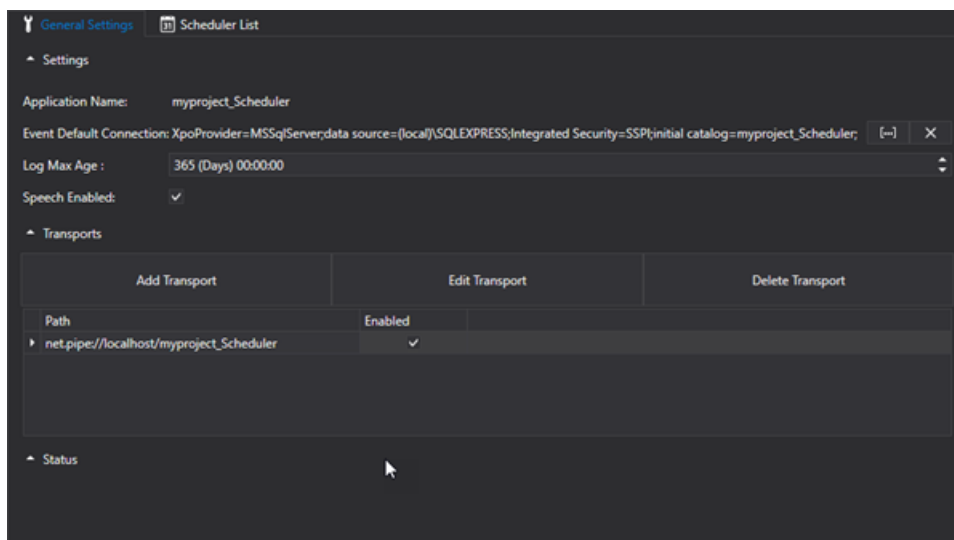
### Stop Server

This command is used for stopping the Scheduler Server manually after the Start command has been used.

## 8.2. General Scheduler Server Settings

The Scheduler resource as a Server has its own Configuration **Settings** section in its workspace. This section is divided into three areas, the main one being dedicated to the Scheduler Server's general Settings.

There are two other smaller sections. One is dedicated to the '**Transports**' settings if the Scheduler Server should ever need to communication with the platform. The other one is dedicated to its 'Status' where the Scheduler Server's system log messages will be shown.



### General Settings

The Scheduler Server's general command settings can be defined and configured in the Settings section:

#### Application Name

The application name is entered here. This is the name with which the OPC UA Server Alarm Dispatcher will be proposed on the Client browser list.

### Log DB Connection

This is the connection string that will be used by the scheduler to record historical event data. If a connection string has been not defined, one will be entered for default when opening the editor. The default connection will be created in the SQL Server database using the local sql server instance and the Application Name as the name of the database.

### Enable Log

Enabling this will generate the log file.

### Log Max. Age

This indicated the maximum age of historical data logged in the Database. Data older than the value set will be progressively deleted.

### Speech Enabled

When this option is enabled, the scheduler events will be vocally enunciated during their execution at runtime.

### Manufacturer Name (Advanced)

This is used to define the name of the manufacturer. This information will be published by the Server.

### Product Name (Advanced)

This is used to enter the product's name. This information will be published by the Server.

### Software Version (Advanced)

This is used to define the software product version. This information will be published by the Server.

### Build No. (Advanced)

This is used to enter the build number.

## Transports

This window is used to define the protocols that are to be used for Server and Client communication. Once the Server has been started up, it will provide end-points according to the transports enabled in the project.



Attention! The "net.pipe" transport is a local transport which required both Client and Server to be started up in the same work session. This transport cannot be used if the Client project and Server project run on different machines.

## Status

This is a log window which activates when the Server is started up manually using the "Start Server" command and reports Server status or error messages.

## 8.3. Advanced Scheduler Settings

### Execution

#### Enable Diagnostics

This allows you to enable several diagnostic functions concerning the server and/or communication drivers.

For further information on Execution Settings, please refer to the Data Server chapter on "Execution".

### Database Settings

#### Default Event Log connection

This is the connection string which is used by the Alarm Dispatcher to record historical data. When the editor opens, a default connection string will be entered if one has not already been defined. The default connection will be created in the SQL Server database using the local sql server's instance and the Application Name as the database's name.

If the connection to the database should disconnect for whatever reason, the log data will be loaded into xml files in the project's "..\AlarmDispatcher\Historian" folder.

#### Max. Event Age

This is used to set the maximum age for the data on DB.

#### Max. Concurrent Historian Accesses (Advanced)

This is used to set the maximum number of concurrent accesses to the server's historian side. Once this limit has been exceeded, subsequent access requests will be suspended until one of the accesses in progress has concluded.

#### Historian Retry Waiting Time (Advanced)

This is used to set the time in seconds to wait between one DB connection attempt and the next.

#### Max. Historian Retries (Advanced)

This is used to set the maximum number of historian access retries before flushing data in the cache to a local text file.

#### Max. Records in Error (Advanced)

This is used to set the max. number of records in error in the cache. Once this limit is exceeded, all records pending a new database entry attempt will be flushed to a text file in XML format in the local disk where the server is running.

#### Min. Records in Cache (Advanced)

This is used to set the minimum number of records in the cache pending database entry, below which will be flushed to a local text file in XML format.

#### Max. Records in Cache (Advanced)

This is used to set the maximum number of records in the cache pending database entry. Once this limit has been reached, the new records will be entered in the cache predisposed to flush data to local text files in XML format. In the event that this limit is reached in the second cache, the records will be irretrievably lost.

**Max. Record Deletions (Advanced )**

This is used to set the maximum number of records to be deleted from the database upon every maximum database age control check.

**Max. History Deletion Processes (Advanced)**

This is used to set the max. number of historical data deletion processes in the Database to be run simultaneously.

**Max. History Restoration Processes (Advanced)**

This is used to set the max. number of processes to be run simultaneously to restore historical data..

**Max. History Flush File Size (Advanced)**

Maximum size in MBytes of all Flush files. Once this limit has been reached, the oldest files exceeding this value will be permanently removed. This will be traced in the log with this message: "The maximum size of safe files allowed has been exceeded causing the permanent removal of some files from path '{0}' !" "

**Max. Alarm Events (Advanced)**

Sets the maximum number of alarm events to be loaded in the buffer.

**Max. History Continuation Points (Advanced)**

Maximum number of continuation points (as above) used for HistoryRead operations.

**Max. Request Age (Advanced)**

Maximum duration of incoming requests. Take note that the oldest requests are deleted.

## Opzioni OPC UA Server

**Min. Publishing Interval**

The minimum time in minutes to publish data supported by the server.

The server resets subscriptions that have a lower interval than this value. The value returned by the server in a subscription indicates when the server will update the value with the field value.

**Max. Publishing Interval**

Maximum time in minutes to publish data supported by the server.

The server resets subscriptions that have a higher interval than this value. The value returned by the server in a subscription indicates when the server will update the value with the field value.

**Publishing Resolution**

This field indicates the minimum difference in ms between the supported publishing intervals. This is always used to calculate the subscription interval which is returned by the server in respect to the one requested by the client.

**Max. Session Timeout (Advanced)**

This is used to set the maximum time in ms with which a Client session will remain open in the absence of Server communication. It is advised to leave this set with the default value.



#### Min. Session Timeout (Advanced)

This is used to set the minimum time in ms with which a Client session will remain open in the absence of Server communication. It is advised to leave this set with the default value.

#### Max. Session Count (Advanced)

This is used to set the maximum number of client session that can be opened on the server.

#### Max. Browse Continuation Points (Advanced)

This used to set the maximum number of continuation points to be used for browsing operations by clients towards the Server.

The "continuation point" is needed to perform a subsequent request to the server to retrieve remaining information. In fact, the amount of information received by the client might be incomplete due to a limited maximum amount of information that the client can receive for each individual interrogation.

#### Max. Subscription Lifetime (Advanced)

Defines how long subscriptions can remain open without being published by client.

#### Max. Message Queue Size (Advanced)

Defines the maximum number of messages saved in the queue for each subscription.

#### Max. Notification Queue Size (Advanced)

Defines the maximum number of notifications saved in the queue for each monitored item.

#### Max. Notification Per Publish (Advanced)

Defines the maximum number of notifications saved in the queue for each publication.

#### Min. Metadata Sampling Interval (Advanced)

Defines the minimum metadata sampling interval.

## 8.4. Types of Schedulers

The Scheduler objects each have a **Event ON** and **Event OFF** execution which offer the possibility to set the desired value to one of the I/O Data Server's tags. Substantially, each Scheduler object executes its own ON and OFF according to the which **Type** it has been set with. This type may be Every minute, hour, day or certain day, Weekly Plan or Calendar.



The Scheduler's 'Runtime editable' property can be enabled to allow the operator to set time schedules in runtime as well.

### Cyclic Scheduler:

The Cyclic Scheduler has a reoccurring event based on a preset time. The options provided are:

- **Every Minute (ON and OFF within each minute)**
- **Every Hour (ON and OFF within each hour)**
- **Every Day (ON and OFF within each day of the week)**
- **Every specific day of the week (ON and OFF within the specified day of the week)**

## Schedule Plans

The Schedule Plan comes with two options, the weekly plan or the Calender. The events in the Calender can be set to occur on a particular prefixed date and time, or a time range comprised of two dates:

- **Weekly Plan**
- **Calender: Prefixed Date**
- **Calender: Day comprised of two fixed dates**
- **Calender: Reoccurring day annually**

For further information please refer to this object's property settings.

## Event ON - OFF in Cyclic Scheduler "Every ... "

When selecting one of the "Every Minute", "Every Hour" or "Every Day" settings, the event will be executed according to the time set in the "**Time ON**" and "**Time OFF**", fields that must be set appropriately to establish the command's ON and OFF execution time within the period indicated.



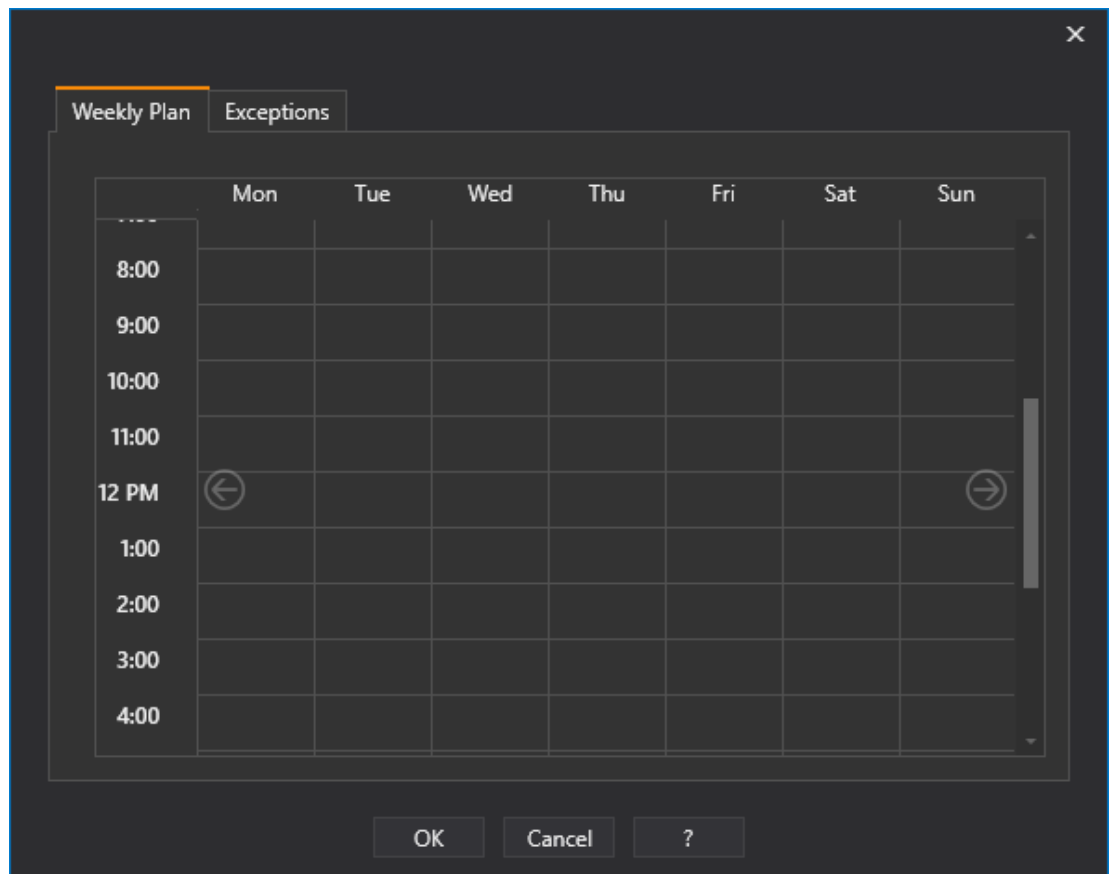
In cases where the "Time On" and "Time Off" values are the same, only the event ON will be executed!

- **everyMinute:** Event executed every **Minute**, with ON and OFF in the **Seconds** field for the set ON and OFF time.
- **everyHour:** Event executed every **Hour**, with ON and OFF in the **Minute and seconds** field for the set ON and OFF time.
- **everyDay:** Event executed every **Day**, with ON and OFF in the **Hour:Minutes:Seconds** field for the set ON and OFF time.
- **everySunday:** Event executed every **Sunday**, with ON and OFF in the **Hour:Minutes:Seconds** field for the set ON and OFF time.
- **everyMonday:** Event executed every **Monday**, as above.
- **everyTuesday:** Event executed every **Tuesday**, as above.
- **everyWednesday:** Event executed every **Wednesday**, as above.
- **everyThursday:** Event executed every **Thursday**, as above.
- **everyFriday:** Event executed every **Friday**, as above.
- **everySaturday:** Event executed every **Saturday**, as above.

## Weekly Planned Events

The Weekly Plan can be selected for planning events through out the week. This is done by using the **Weekly Plan** settings that can be accessed using the Tab in the settings window.

The Weekly Plan's settings window provides a graphical user interface grid showing a week divided into days and hours. This window is used to configure the activation time or time range of the command to be scheduled throughout the week.



*This is a Weekly Plan window used to program events at certain times on a weekly basis*

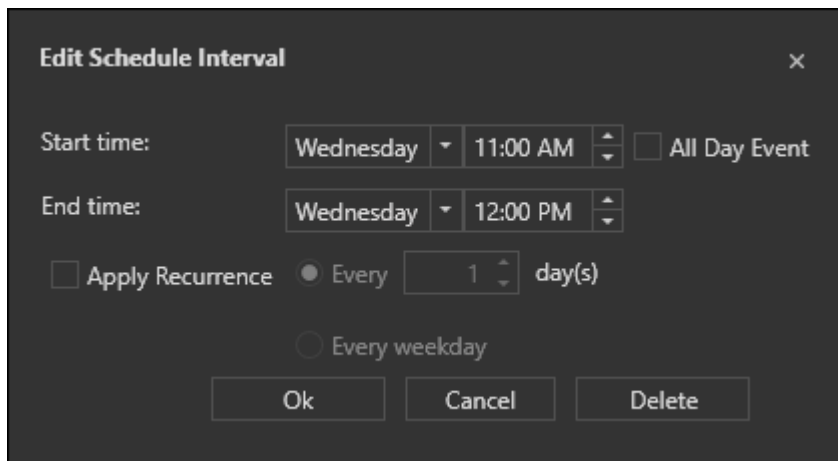
Different command activation time ranges can be set throughout in the Weekly Plan including several on the same day. The minimum time range settable is a minute. The ON event will be executed on the '00' second of the time range's start minute. The OFF event will be executed on the time range's '59th second of the last minute.



**CAUTION!** In cases where the pc's clock is turned back, and the time set in the "weekly Plan" is changed to coincide with the pc's new clock, the scheduler command will not be executed.

### Inserting Times

To insert a new time range simple double click on the box of the day and hour desired or right click and activate the "Add Schedule Time" to open a window where a new time range can be entered.



**Edit Schedule Interval** [X]

Start time: Wednesday 11:00 AM ☐ All Day Event

End time: Wednesday 12:00 PM

☐ Apply Recurrence ☒ Every 1 day(s)

☐ Every weekday

Ok Cancel Delete

*Window used to insert or modify time ranges*

The above window is used to insert a time range by defining the start and end times and the day or days of the week in which the time range is to start and finish. If two different days are entered in the 'Day of Week' fields, the time range will start with the first day and end with the second. The time ranges can also be dragged and repositioned with the mouse.

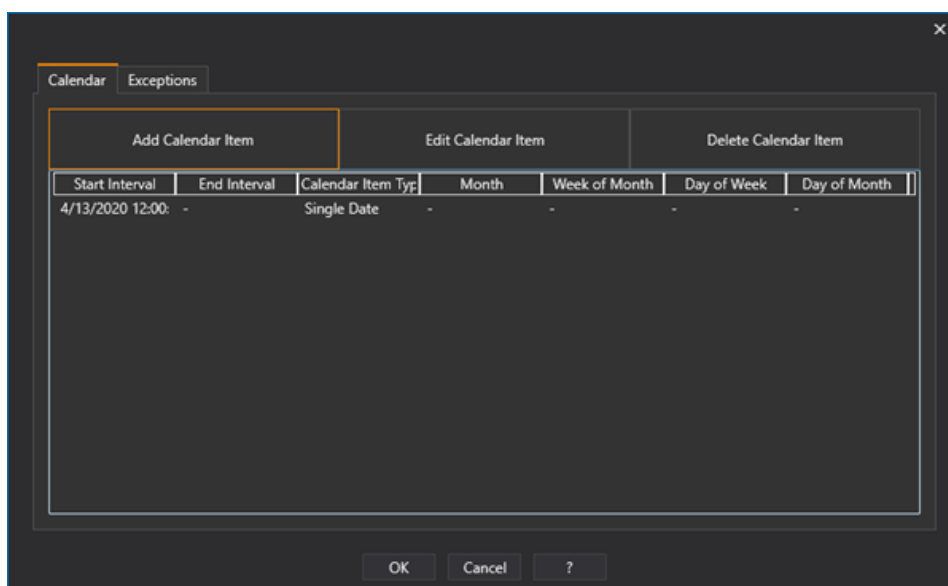


The events in the Weekly Plan View will be shown in calendar format in WebClient.

## Scheduled Calendar Events

When selecting "Calendar" event type, the events will be scheduled based on specific **Calendar dates**. In order to set these dates you will need to use the **Calendar** settings which can be accessed through Tab shown in the settings window.

The Calendar settings window has a user interface that allows you to insert, modify or remove each single Calendar Item type inserted. Each inserted Calendar Item Type must then be set appropriately through the same window.



Calendar Exceptions

Add Calendar Item Edit Calendar Item Delete Calendar Item

Start Interval	End Interval	Calendar Item Type	Month	Week of Month	Day of Week	Day of Month
4/13/2020 12:00:	-	Single Date	-	-	-	-

OK Cancel ?

*The Window used to set Calendar events.*

Different event types can be inserted by using the "Add Calendar Item". The different events are:

### Single Date

When selecting a "Single Date" event you will be able to define a fixed date and time for command execution. The ON event will be executed according to the date and time set and the OFF event will be executed at the beginning of the following minute.

### Data Range

When selecting this event type, it will be possible to set its start and end date. In this case, the ON event will be executed according to the Start Date and Time set and the OFF event will be executed according to the End Date and time set.

### WeekNDate

When selecting an "WeekNDate" event it will be possible to define a plan based fixed on days, weeks or months. The field provided to enable this are:

- **Month:** this field is used for selecting the month in which to execute the even. Specific individual months can be selected, or odd months or even months only or all the months of the year.
- **Week of Month:** this field is used for selecting the week in which to execute the event. The first to the last week can be selected or all the weeks of the year.
- **Day of Week:** this field is used for selecting the day on which to execute the event. One specific day or all the days of the week can be selected.
- **Start Time:** this field is used for selecting the date and time in which to execute the ON event.
- **End Time:** this field is used for selecting the date and time in which to execute the OFF event.

## 8.5. Scheduler Properties

Each Scheduler inserted in the project's Scheduler Resource is configured like an 'object' which provides a scheduler with its own autonomous commands in respect to the other eventual schedulers inserted. Each scheduler object can be managed in runtime by the operator and using the Scheduler Viewer object from the toolbox for screens in visualization Movicon.NExT Client.

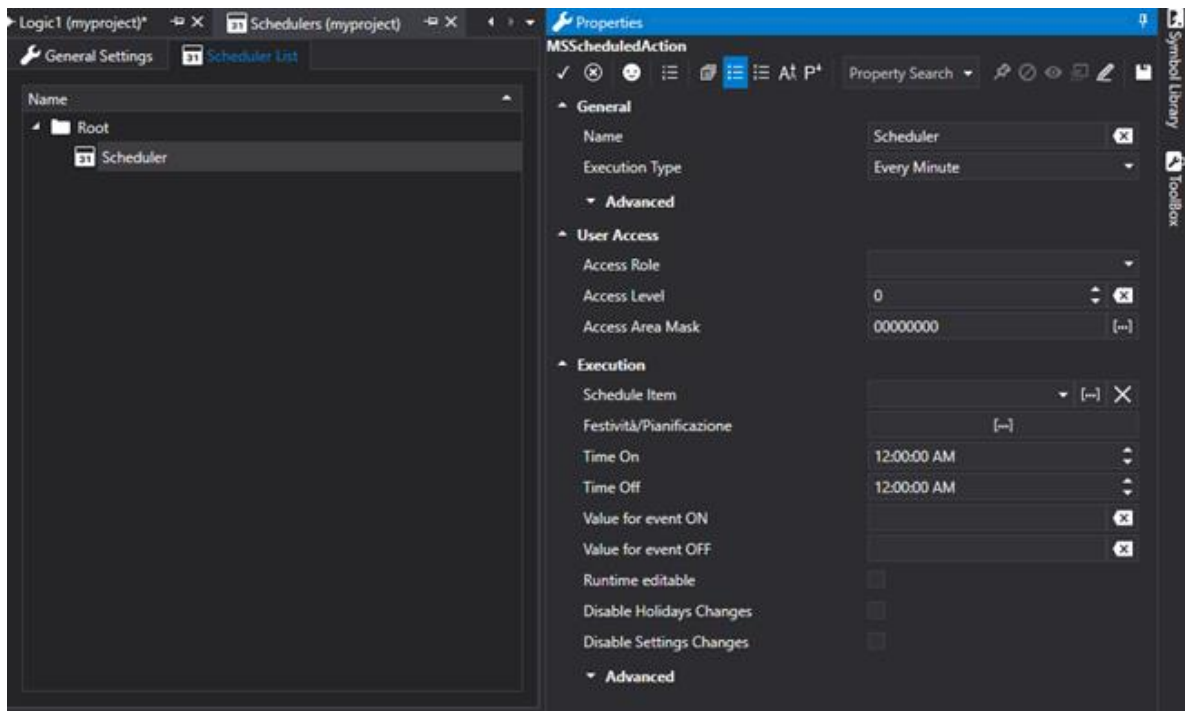
Each Scheduler object is configured when inserted using a settings window or at any other time using its Properties window.

The Scheduler properties are described below.

### Scheduled Event List

A list or predefined scheduled events is reported in the "Settings" section's "Scheduled Event List" window. Through the menu project> schedulers it is then possible to add other events. Events can be organized into folders to be created with the "Add New Folder" command.

When a new event is added, a popup window will open to defined its properties. The window can also be opened by double clicking on an already existing event.



*Window used for setting Scheduled Event properties.*

## General Properties

The general properties of a Scheduler Object are used for define the general configuration settings:

### Name

Represents the name associated to the scheduled event. The name must be unique within each individual folder.

### Execution Type

Represents the type of schedule to be used. The different types that can be used are described in the "Types of Schedulers" topic.

## Advanced

### Enable

This option is used for enabling or disabling the scheduled event. New events are inserted with this option enabled for default.

### Enable Tag

This field is used to select the I/O Data Server Tag in which the scheduler command will be applied.

### Hide Runtime Exception Tab

Enabling this property will prohibit the Holidays Tab from showing at runtime for the scheduler in question and therefore making it impossible to view and edit its list of holidays during runtime.

### Hide Runtime Settings Tab

Enabling this property will prohibit the Settings Tab from showing at runtime for the scheduler in question making it impossible to view and edit its settings during runtime.

## Execution Properties

The execution properties of a Scheduler object are settings that define its functionality:

### Schedule Item

This field is used to insert the name of a Tag to be used for enabling the scheduled event in runtime. When a Tag is specified in this field, the event will be disabled when the tag is set with a 'zero' value and enabled when the tag is set with a different value that is not zero in runtime.

### Holidays/Planning

This command is used to open a window through which you can define holiday dates and periods or specific intervals during which the scheduler will not execute the commands that it has been set with. For example, you can set 25th December as a holiday independently from which year you are in at the moment.

When the holidays window opens for the first time, or each time holidays have not yet been set, a message-box will appear asking if you want to holidays for the current year. Upon confirming this message-box, the public holidays will be automatically set according to the culture defined in the system (for example, Italian public holidays are different from those in the USA). In addition, some holidays vary according to the current year (for example, the days which Easter falls on changes every year). Those public holidays that do not vary, such as Christmas) will be automatically be inserted in "WeekNDate" mode that does not require you to specify a year. Other public holidays will be inserted according to the date's of the current year.

The list of holidays can be changed as pleased by adding, deleting or modifying each item listed. The list of holidays can also be modified at runtime using the Scheduler Window object.

### Time On

This field is used to enter the time when the scheduled On event is to be executed. The complete hour, or part of it such as the seconds or minutes, are only taken into consideration according to the event type selected in the "Type" field.

### Time Off

The time when the scheduler Off event is to be executed is entered in this field. The complete hour, or part of it such as the seconds or minutes only are taken into consideration depending on the event type selected in the "Type" field. If the "Time Off" is equal to the "Time On", the OFF command will not be executed.

### Value for Event On

This field is used for inserting the value to be applied to the Tag when the scheduled ON command is executed.

### Value for Event Off

This field is used for inserting a value to be applied to the Tag when the scheduled OFF command is executed.

### Commands on On

This is used to set a list of commands when event becomes active.

### Commands on Off

This is used to set a list of commands when event becomes inactive



The Scheduler's On and Off commands are launched by the Movicon.NExT Client only when Runtime is completely started up from command line with the "-start" option.

**Attention:** when the Client is launched from command line with the "-start -client" option, the commands will not be executed.

### Runtime Editable

Enabling this option will allow the scheduled event to be modified in runtime by using the "Scheduler Viewer" object. When this option is enabled, any modification in runtime will also be linked to the user authentication fields: "Access Role", "Access Level" and "Access Mask".

### Execute ON Event at Startup

This property is only available for "Weekly Plan" and "Calendar" Scheduler types.

When this property is set to True (default) and if the ON time band is active at the Scheduler startup, the relative command will be executed even when already executed before the Scheduler was arrested.

When the property is set to False, at the Scheduler startup, if the ON time band is active and the relative command already executed before arresting the Scheduler, this command will not be repeated.

This property is editable at runtime.

### Execute OFF Event at Startup

This property is only available for "Weekly Plan" and "Calendar" Scheduler types.

When this property is set to True (default) and if the OFF time band is active at the Scheduler startup, the relative command will be executed even when already executed before the Scheduler was arrested.

When the property is set to False, at the Scheduler startup, if the OFF time band is active and the relative command already executed before arresting the Scheduler, this command will not be repeated.

This property is editable at runtime.

## User Access

The User Management properties of a Scheduler object define its security settings:

### Access Role

This field is used for selecting a user "Role" needed for performing scheduled event runtime modifications. In this case only the users assigned this role can modify the scheduler. The settings of this parameter are considered only when the "Runtime Editable" option has been enabled.

### Access Level

This field is used for selecting a user "level" needed for performing scheduled event runtime modifications. In this way only those users with levels equal to or higher than the one set in this field can modify the scheduler. The settings in this parameter will only be considered when the "Runtime Editable" option has been enabled.

### Access Area Mask

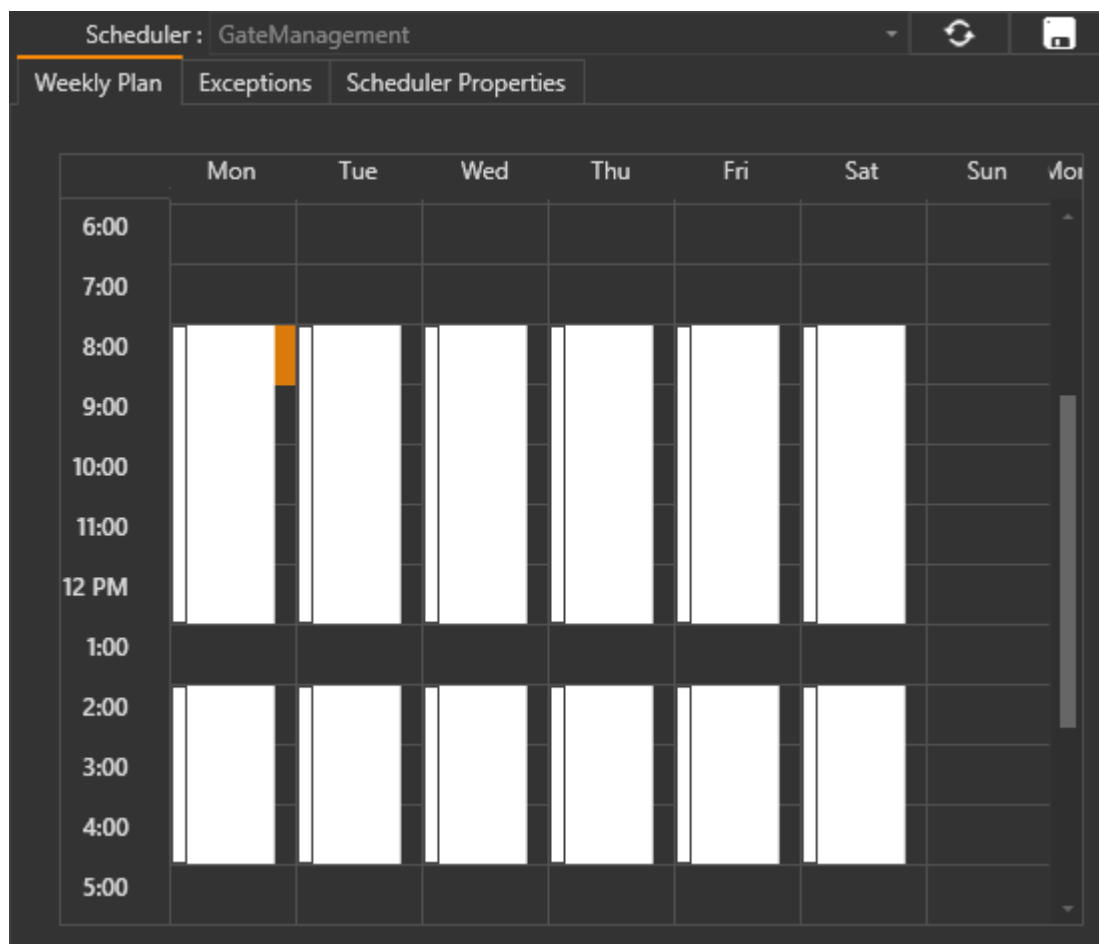
This field is used for selecting a user "Mask" needed for performing scheduled event runtime modifications. In this case only those users with access masks that are compatible to the one set can modify the scheduler. The settings in this parameter are only considered when the "Runtime Editable" option has been enabled.



## 8.6. The Scheduler Viewer Object

The Scheduler Server can be displayed and edited by the operator during runtime by using the graphical Scheduler Viewer object located in the Viewer category in the screen toolbox.

The Scheduler Viewer object can be inserted on screen and configured as pleased through its properties as described in the topic relating to Graphical Screen displays.



*This is an example of the Scheduler Viewer object that can be inserted on screen.*

By using this graphical object, the operator can set scheduled events - where consented in the properties by the programmer - per each scheduler server object. The settings entered by the operator in runtime will have priority over those times set in object's properties in edit mode.



# 9. Scripts VB.NET

## 9.1. Introduction

The programming guide contains the information you need to develop and create Scripts in the project using the WinWrap® Basic software component by Polar Engineering integrated in Movicon.NExT.

The main characteristics of WinWrap® Basic are:

- permits you to create Script in WWB.NET™ language, compatible with Microsoft Visual Basic.NET™
- permits to extend the set of instructions with custom functions and methods
- Consents to insert references to external .NET assemblies

### Limitations

Sine the WWB.NET™ language is compatible with VB.NET it presents the following limitations:

- The timers from the System class. As threads are not supported, the timers from the System.Timers class can be used instead.
- Only one thread can be run for each script.
- Type inference is not supported.
- Anonymous methods are not supported
- LINQ is not supported.
- The XML literal expression are not supported.
- Methods implemented by modules / macros cannot be overloaded.
- The classes implemented by WWB.NET are not .NET classes.
- The generic calls and methods are not supported (generic types are supported)
- The string variables cannot be set with Nothing and are never Nothing,
- The {array} syntax is only supported for one-dimensional arrays.
- The <attribute> syntax is allowed but ignored.
- The following VB.NET keywords are recognized but not supported: FileGet, FileGetObject, FilePut, FilePutObject, Inherits, MyBase, MyClass, MustInherit, MustOverride, Namespace, Narrowing, NotInheritable, NotOverridable, Operator, Overloads, Overrides, Share, SyncLock, Widening.

### Unicode support

WinWrap® Basic accepts the use of stings in Unicode format to permitting:

- Unicode strings to be displayed in the Scripts dialog window.
- texts to be managed in Unicode format in the project's string variables.
- UTF8 or UTF16 Unicode files to be read and/or written using the"vbUTF8BOM" and "vbUTF16BOM" constants. Writing one of these constants in the first character of a test file will determine its encoding.

### Limited Warranty

Attention: The use of script code in Movicon projects is subject to LIMITED WARRANTY. The user has in fact the freedom to manage scripts to utilize events or methods, both of the product or of third parties. Progea may not be able to fully guarantee that the product will behave correctly in cases of script code using certain properties, events or methods. The user should always refer to documentation whenever available and eventually make preventive checks on the correct functioning of the code they

intend to use. In any case, Progea provides LIMITED WARRANTY on the project's functionalities using VB.NET script. If you need any help or further information please contact the Progea Technical Support Services.

## Copyright

All the information contained in the Movicon.NExT documentation presupposes:

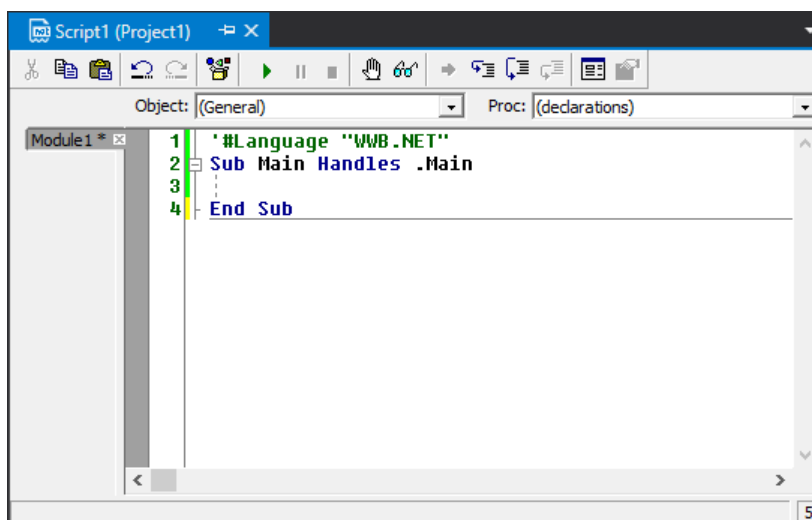
- Windows refers to the registered Microsoft inc. trademark for versions Windows 7, Windows 8 and Windows 10
- Movicon.NExT refers to the supervision system developed by Progea protected by the international laws on Copyrights.
- WWB.NET refers to the WinWrap® Basic component of Polar Engineering integrated in Movicon.NExT, compatible with the Microsoft Visual Basic.NET language.
- Ms Access and SQL Server refers to the products registered by Microsoft inc.
- Any other cited product or brand name is registered or protected by their relative owners.

## 9.2. Script Editor

The script editor is composed of a window containing:

- a toolbar
- the "Object" and "Proc" combo boxes
- an area with a white background within which code is edited
- a contextual menu

The structure of this window is the same for editing a Script Resource and editing a Script of an Object/Screen Page or Tag.



## Toolbar

The Script editor toolbar provides some useful tools to run and debug the script (as explained in the 'Basic Script Debug' topic).



### Browse



This command is used to access the list of functions and properties available in the basic script. The window which opens is called function Browse.

### Start/Resume



This command starts the basic script execution. When the code is referred to the project's basic resource, the Sub Main contents are executed. Otherwise, if it is referred to an object, it enables the object's events management. It is used to carry out a syntactic check of the script code, in cases where it is not referred to a project resource which require the project to be run (e.g. read variables), the code itself is also executed (e.g. reading a value from DB).

### Pause



This command pauses the basic script that is running by positioning itself and coloring the line of code being executed in yellow.

### End



This command stops the basic script from being run. The code must be in pause for this command to function.

### Break Point



This command inserts or removes a break point on the row in which the cursor is present. This function is available with a click on the extreme left border of the editor window coinciding with the row where you wish to insert the break point or with the F9 key. The moment in which the basic script must execute a line of code in which a break point is present, the execution is automatically put in pause.

### Quick Watch



This command returns the result of the highlighted function in a dialog window or in the basic script's debug window (Watch Window).

### Show Current Statement



This command is used to position the cursor to the next instruction which must be executed during the routine's debug phase.

### Step Into



This command executes a function step-by-step each time the command is pressed.

### Step Over



This command executes a function step-by-step each time the command is pressed.

### Step Out



This command executes the functions up to the line indicated by the cursor.

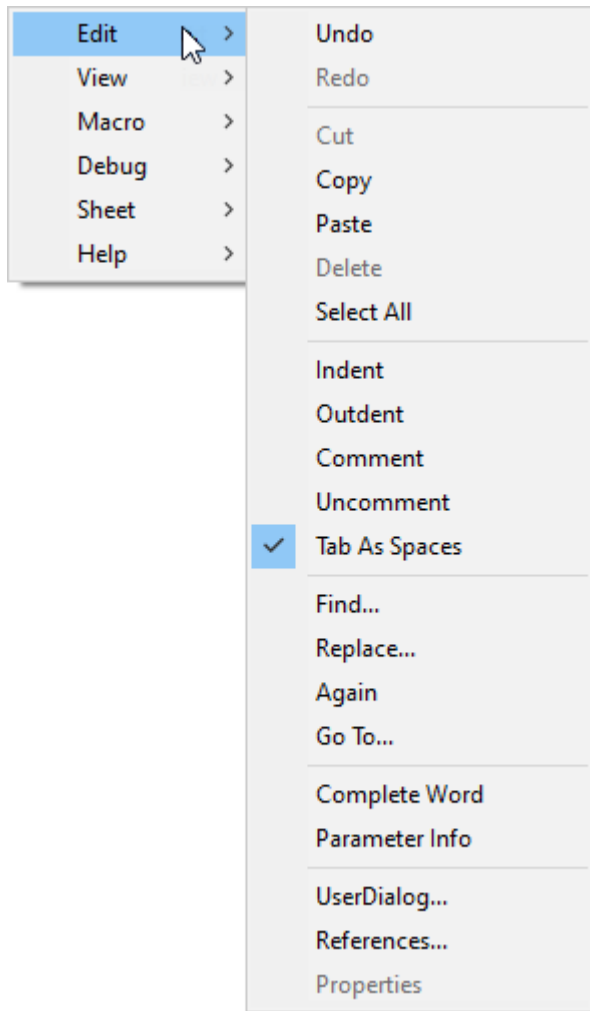
## Combo-box Object and Proc

There are two combo-boxes underneath the toolbar: Object e Proc.

The objects defined in Movicon.NExT to which the script refers to are displayed in the "Object" list. The selection of one object in respect to another on the Object list will change the list of procedures in the "Proc" list which are available for that object selected.

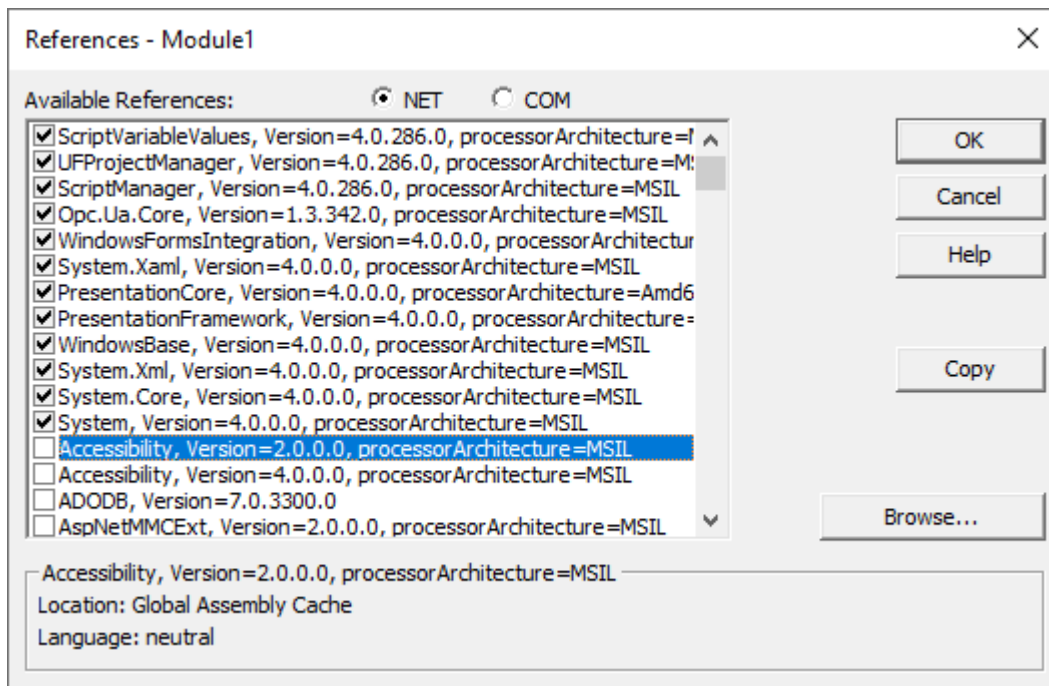
## Contextual Menu

There is a contextual menu like the one shown below in the area where script code can be edited:



Among the various items you will find these commands:

- "Macro > Check Syntax": this allows you to carry out a check on the script's syntax
- "Edit > Replace": this gives you access to the Find/Replace tool
- "Edit > References...": this opens a window containing the References to which Movicon.NExT assembly references can be added (e.g. objects from the Toolbox) or external assemblies as long as they are in .NET language



### 9.3. Basic Script Debug

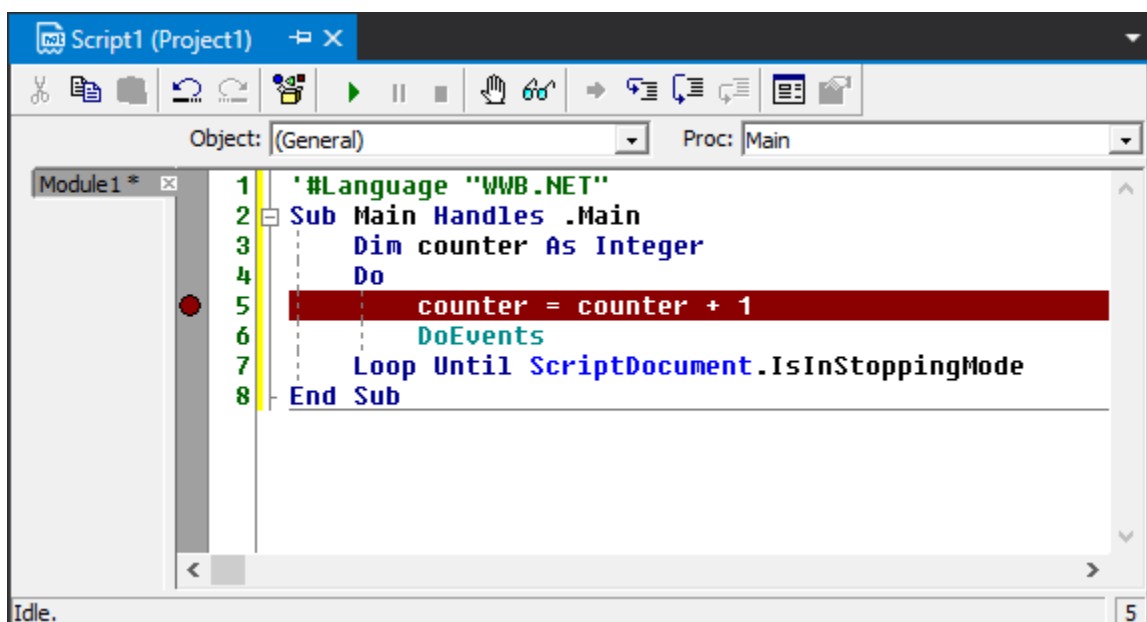
Movicon.NEXT allows complete debugging of the project's scripts, be they Resource Script or Script within Graphic Objects or within synoptics.

Debugging can take place within the Editor or during project execution in Runtime.

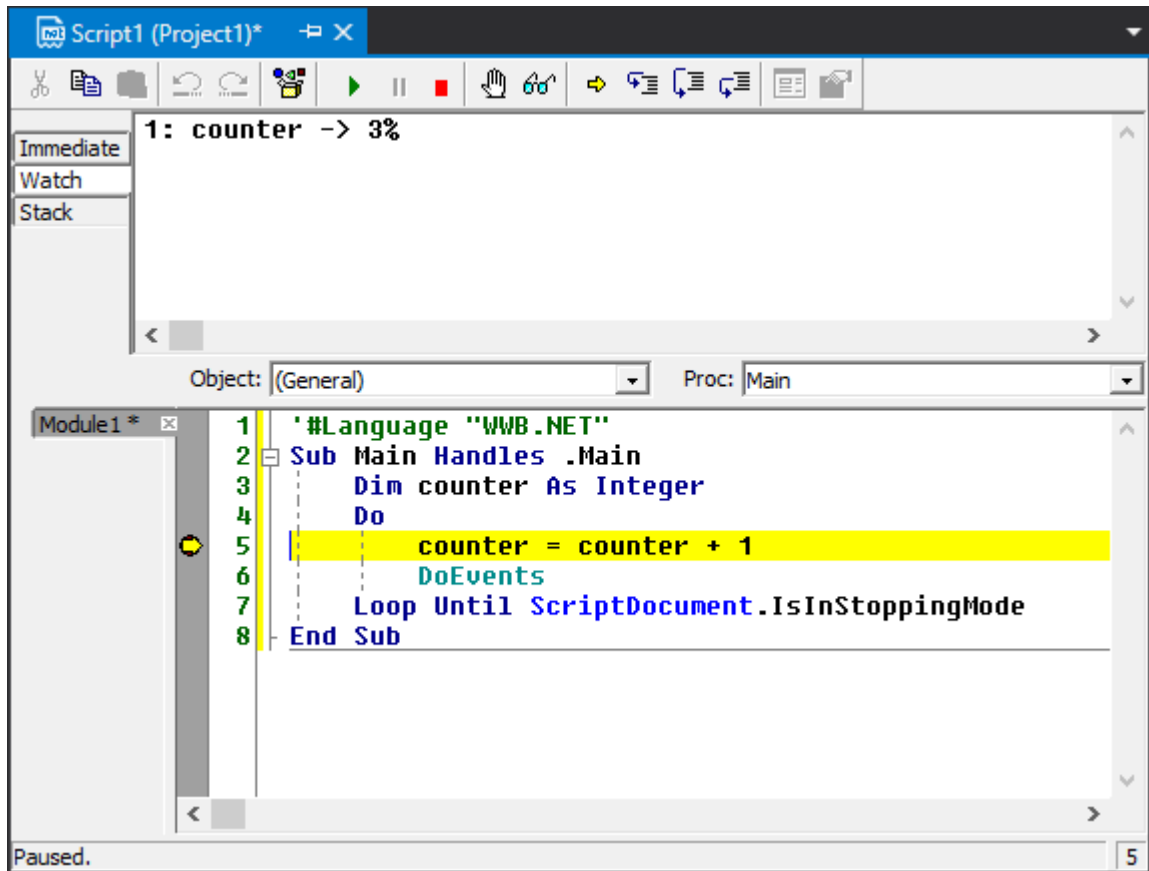
During the editing of the project it is only possible to debug the Scripts defined as Project Resources by launching their execution with the play button of the toolbar, while it is not possible to test the scripts associated with the objects or synoptics as these scripts are managed as an event during Runtime.

While running the project in Runtime, however, you can debug all scripts.

To debug and proceed with a step-by-step analysis of the code, during programming it is possible to define breakpoints as shown in the following image:



When the script is executed and arrives at the instruction associated with the breakpoint, the debug window is opened containing, in the upper part, the items "Immediate", "Watch" and "Stack" and in the lower part the code in execution with highlighted the line on which execution was stopped.



- **Immediate:** allows you to view the output of any debug.print and allows you to set a value of a script variable.  
For example, to set the value 10 to the counter variable, simply write "counter = 10"
- **Watch:** allows you to monitor the value of script variables. To monitor the variables, simply write the name of the variable and confirm with Enter, or drag the name of the variable into the watch window by selecting it from the script code.
- **Stack:** lists the lines that called the current statement. The last line is the current statement. The first line is the first statement executed.

Once the breakpoint line has been reached, you can continue step-by-step, or resume execution with the toolbar commands.



The insertion of breakpoints inside the scripts is permanent, as they are saved in the properties of the script, and therefore they will also be active in Runtime mode. It is therefore recommended to remove the breakpoints after completing the debugging phase, before running the project in production.



Some script functions performed by the editor in debug can return with values compared to the execution in Runtime, such as reading the value of a Tag. It is recommended to use programming debugging only to roughly verify the code and repeat the test accurately by launching the project.



## 9.4. Basic Script Variables

PlatformNExT Address Space variables and local Basic Script variables can be used in the Platform.NExT Basic Scripts. The Address Space tags have globalized use the project and the local Basic Script variables are only visible within Basic routines and are destroyed once the Basic Script has terminated. In addition it is also possible to use the Client's "Local" and "System" variables.

The project variables (Platform.NExT Address Space) can be directly used with their name, or by using the "ScriptDocument.GetVariableValue()", "ScriptDocument.SetVariableValue()" functions with a Script resource, or with the corresponding "Document.GetVariableValue()", "Document.SetVariableValue()" functions for code edited at screen object level.



The variables internal Basic Scripts have priority over those of the project. Meaning that when using the name of a project variable directly in the basic script code where a variable already exists with the same name, this will be ignored and the already existing one will be set or read. In this case you will have to use the GetVariableValue(), SetVariableValue() basic functions.

Bit type variables, when used directly with their name, are converted into boolean (true or false). This means that a project variable with value "1" will be read as a "true" variable inside the Basic Script and therefore with a numeric value equal to "-1". The "0" value is interpreted as "false" by the Basic Script which will always be a numeric value equal to "0". This mechanism also goes for both the reading and writing of bit variables inside Basic Scripts. The GetVariableValue() and SetVariableValue() basic script functions can always be used to obviate this mechanism. In this way the function's return value will always be the numeric value of the bit, "1" and "0", and not the boolean value.

Example::

When reading the bTest variable declared in the Platform.NExT DataBase as bit and set to "1", the result will be:

```
Sub Main Handles .Main
    Debug.Print bProva 'Risultato = -1
    Debug.Print ScriptDocument.GetVariableValue("bProva") 'Risultato = 1
End Sub
```

### Variables without sign

The WinWrap does not provide the use of variables without sign. Therefore in order to use a Address Space WORD (without sign) type or DWORD (without sign) variable type in the Basic Script it must be converted to avoid causing overflow errors. Overflow error occur when the variable used in the script exceeds the INT type value (ie. 32767 for one Word) because the basic script engine does not manage variables without sign. **To void this problem a UInt32 variable type is always passed to the WinWrap as Long.**

In cases for variables contained inside a folder, for example if there is a "Var1" variable in the "F1" folder, the direct notation from script to refer to this variable would be:

"F1\_Var1"

Based on the script syntax, it isn't possible to differentiate the variables contained in a folder from those that have names beginning with the same name of the folder.

For example "F1\Var1" and "F1\_Var1" would both be referred with the direct "F1\_Var1" notation.

Therefore great care must be taken to avoid creating overlaps such as these or otherwise indirect notation should be used to avoid misinterpretation:

**ScriptDocument("F1\Var1")** for the variable inside the folder and **ScriptDocument("F1\_Var1")** for the variable in the root.

## Array Variables

In cases where Array variable types are used, each array element can be accessed by specifying its number between rounded brackets by using direct tag access syntax. For example, let's suppose that we have created the 'var\_01' array tag type in the server's address space, in order to access the tag's elements we would use the following syntax:

```
Var_01(0) = 1  
Var_01(1) = 2
```

However, direct syntaxes can only be used in script on the Client side and not in that of the Server side. Despite this, it is also possible to access Array elements using script functions with the following syntax:

### Script on Client side:

```
ScriptDocument.GetVariableValue("Var_01")(0) = 1  
ScriptDocument.GetVariableValue("Var_01")(1) = 2
```

### Script on Server side:

```
ServerScriptManager.GetTag("Var_01").Value(0) = 1  
ServerScriptManager.GetTag("Var_01").Value(1) = 2
```

However, please note that even though the script functions are used to write an Array element, it will not be notified to the Client which will continue showing the old value. In order for the Server to notify the Client of this change, the entire Array variable will have to be set. Therefore, you will need to use a temporary variable within the script as in the following example where "Folder1\Tag\_ArrayFloat" is the Server's Array variable:

```
'#Language "WWB.NET"  
Option Explicit  
Sub Main Handles .Main  
    Dim myArr() As Single  
    Dim nLen As Integer  
    Dim iCounter As Integer  
    nLen = UBound(Folder1_Tag_ArrayFloat)  
    ReDim myArr(nLen)  
    myArr = Folder1_Tag_ArrayFloat  
    For iCounter = LBound(myArr) To UBound(myArr)  
        myArr(iCounter) = myArr(iCounter) + 1  
    Next iCounter  
    Folder1_Tag_ArrayFloat = myArr  
End Sub
```

## Structure Variables

When using structure variables directly, the delimiter character to use between the name of the variable and the member is the dot ("."). When using the GetVariableValue() and SetVariableValue() functions instead, the separation character to use between the variable name and the member is the colon (":"). For example, if we define the "TagStruct01" structure type variable with a prototype which has "Member\_1" and "Member\_2" members, to access the variable the syntax will be:

- Direct access (Client side script only): TagStruct01.Member\_1 = 1

- Indirect access (by means of Client side functions):  
`ScriptDocument.SetVariableValue("TagStruct01:Member_1", 1)`
- Indirect access (by means of Server side functions):  
`ServerScriptManager.GetTag("TagStruct01:Member_1").Value = 1`



Attention! When using the direct syntax, only the simple members defined in the prototype's root can be accessed and not those members defined in the prototype's folder or membered defined as prototype (nested structures). In this case, it will be necessary to use the script function to access nested members.

For example, if we define a "TagStruct01" variable structure type with a prototype which has a "Member\_1" member in the prototype's Folder1, the syntax to access the variable will be:

```
ScriptDocument.SetVariableValue("TagStruct01:Folder1\Member_1", 1)
ServerScriptManager.GetTag("TagStruct01:Folder1\Member_1").Value = 1
```

Similarly, when having defined the "TagStruct01" variable structure type with a prototype that has a "MemberStruct" member that has a "Member\_1" member in the nested prototype's Folder1, the syntax to access the variable will be:

```
ScriptDocument.SetVariableValue("TagStruct01: MemberStruct \Folder1\Member_1", 1)
ServerScriptManager.GetTag("TagStruct01: MemberStruct \Folder1\Member_1").Value = 1
```

## Variables inside Folders

In cases accessing variables contained inside folders, the "\_" character must be used as a folder and variable separator or folder and folder separator. When using the `GetVariableValue()` and `SetVariableValue()` functions to access variables in folders the "\" character must be used as the separator. For example, when accessing the "Var1" variable inside the "Folder1" Folder, the following syntax must be used:

- Direct access:  
`Folder1_Var1 = 1`
- Indirect Access (by use of functions)  
`ScriptDocument.SetVariableValue("Folder1_Var1", 1)`

Or with more folder levels:

- Direct access:  
`Folder1_Folder2_Folder3_Var1 = 1`
- Indirect access (by use of functions)  
`ScriptDocument.SetVariableValue("Folder1\Folder2\Folder3\Var1", 1)`

When accessing tags directly you need to make sure that you don't define the names of folders and variables that might result as homonyms. For example, If you define "Motor1" variable inside the "Machine1" folder in the root of the address space, and a second variable directly in the address space's root as "Machine1\_Motor1", the script intellisense will have problems distinguishing them from each other because the syntax you would have to use to access them both would be the same: "Machine1\_Motor1". In situations like this, it would be better to use the `GetVariableValue()` and `SetVariableValue()` in order to make distinctions between the two tags.

## TimeStamp and Quality

Variable TimeStamps and Qualities can be read using script by using direct "VarName\_Quality" and "VarName\_TimeStamp" notation.

Avoid creating variables with the same name of existing variables with the "\_quality" and "\_timestamp" suffixes as it may be difficult to distinguish whether the quality of the variable or the variable with that name is to be referred to. The "GetVariableQuality" and "GetVariableTimeStamp" functions can also be used to refer to the variable's quality and timestamp as an alternative to distinguish the differences as described above.

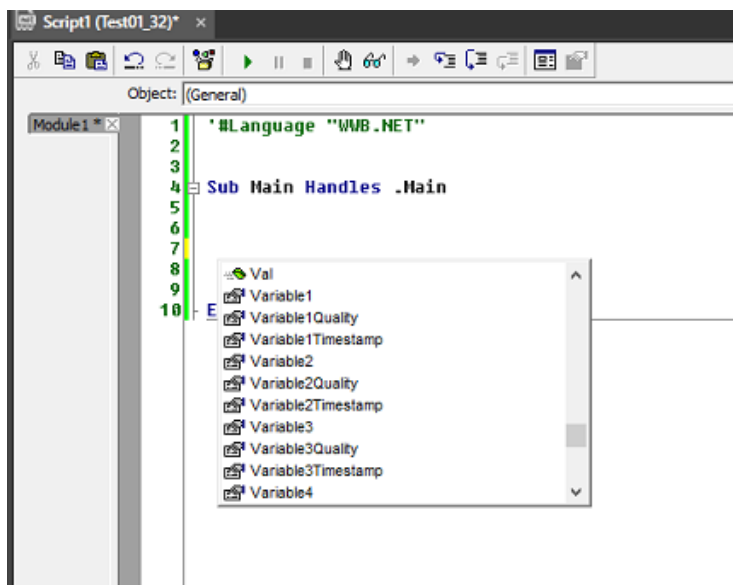
## Client system and Local variables

The Client's local and system Tags can be accessed by script code. To access local tags digit "TemporaryVariables" followed by a dot to open the list of selectable members. To access system tags digit "SystemVariables" followed by a dot to open the list of selectable members. For example:

- TemporaryVariables.LocalTag1
- SystemVariables.CurrentUser

## Intellisense in Scripts

The Basic Script resources are able to identify the project's variables when written directly and in those cases where the tag name is written in blue. This comes in handy when using the "CTRL+Space Bar" to open the list of available methods and properties as well as the project's tags where you will be able to select the name of a tag instead of having to write it. Therefore when you press the "CTRL+Space Bar" within the script, a list box will appear like the one below allowing you to scroll and select the tag desired:



## 9.5. Basic Script Types

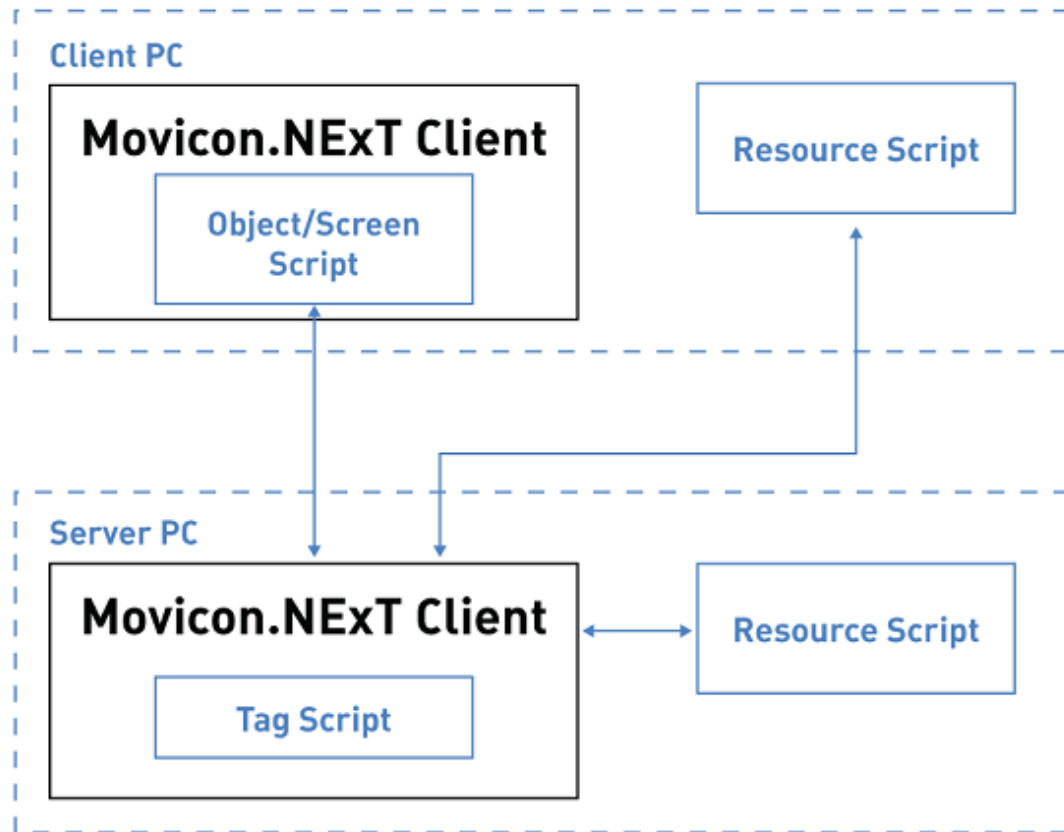
### 9.5.1. Script Typology

The Movicon.NExT scripts can be developed in three different project areas:

- Script Resource: these are the scripts that are defined in the Project Explorer and can be executing using a command or at project startup.

- **Scripts in Objects/Screens:** these are the scripts associated to the graphic objects (User Controls or Symbols) that are present in screens or associated directly to screens.
- **Scripts in Tags:** These are the scripts defined in the Tags from the I/O Data Server.

According to the area they belong to, the scripts can be executed on Client or Server side as shown below in the diagram:

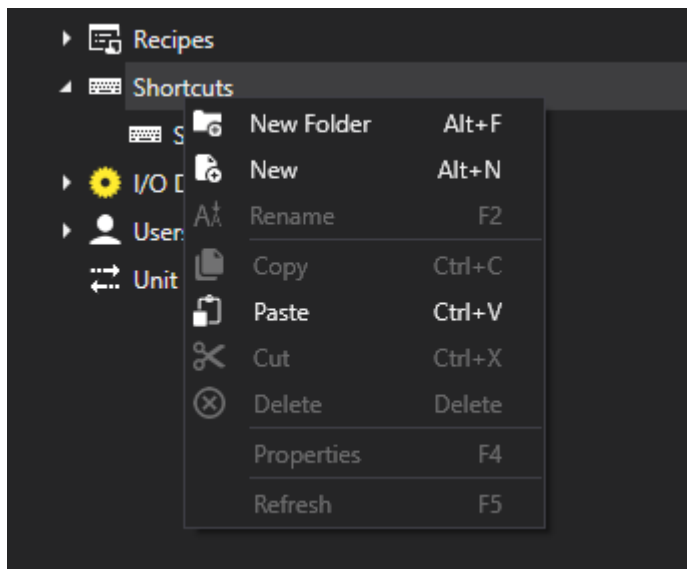


Basically:

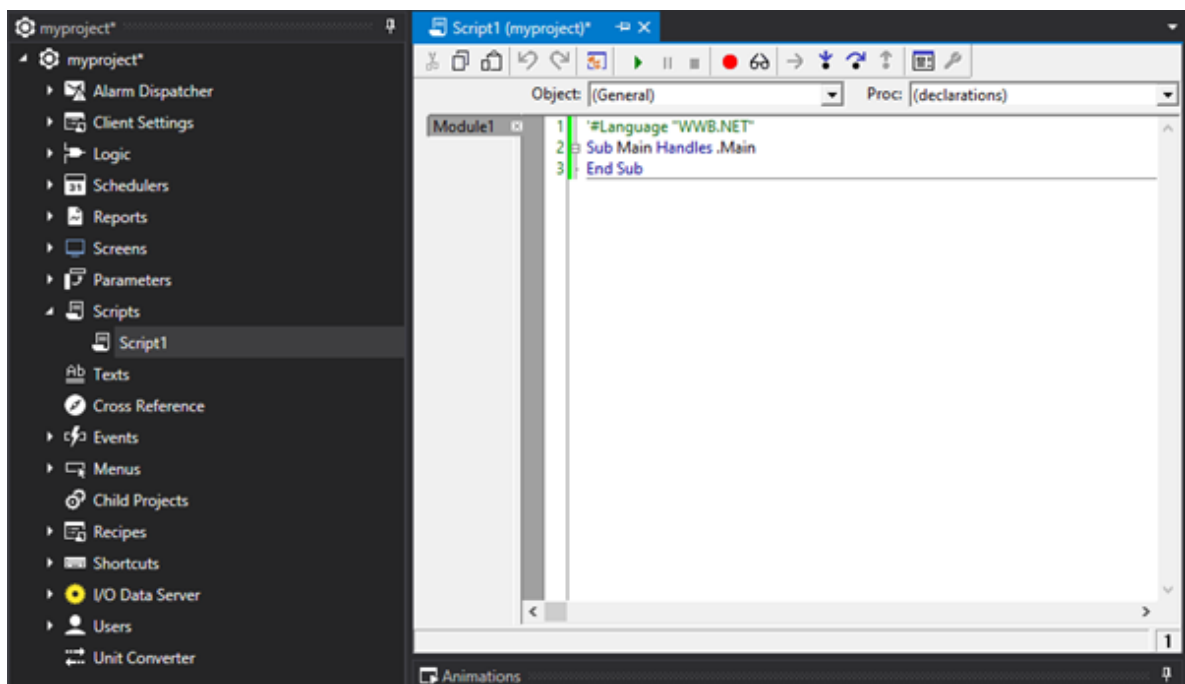
- **Scripts in Tags:** are executed within the project's I/O Data Server. Scripts of this type do not depend on the presence of one or more Clients as the code execution is centralized within the I/O Data Server.
- **Scripts in Objects/Screens:** are executed within the project's Client, being in the part of the project's graphical interface. When there are more than one Client, each Client executes the scripts associated to the objects or to the screen pages which it is displaying.
- **Script Resource:** scripts from the resource always interface as Client towards the project's I/O Data Server but they can be executed in the PC where the Client is running or on the PC where the I/O Data Server is running using the "Run On Server" option (with the possibility to run them by means of the Script Service). In applications with Clients, it is possible to choose where to run the Script Resource, for example, by centralizing the execution of a script that generates a report and make the results available to all clients, or locally on each client, for example to execute a script that opens the alarm screen when the alarm is activated.

### 9.5.2. Script Resource

To add a Script Resource simply select the "Scripts" item from the "Project Explorer" window then click the "New" button from the contextual menu that appears.



When confirming the operation, the new Script Resource will appear within the Project Explorer with the editor window of the Script code just entered open. The Script can subsequently be renamed by always using the contextual menu.



A Script Resource contains the Sub Main procedure in which you must enter the instructions to be run the moment in which the script is called. Once the subroutine has terminated, in the absence of a programmed loop cycle, the script is terminated ready for the next call.

There are two different modes to use to start running a Script Resource:

- on command
- at project startup

## Run on command

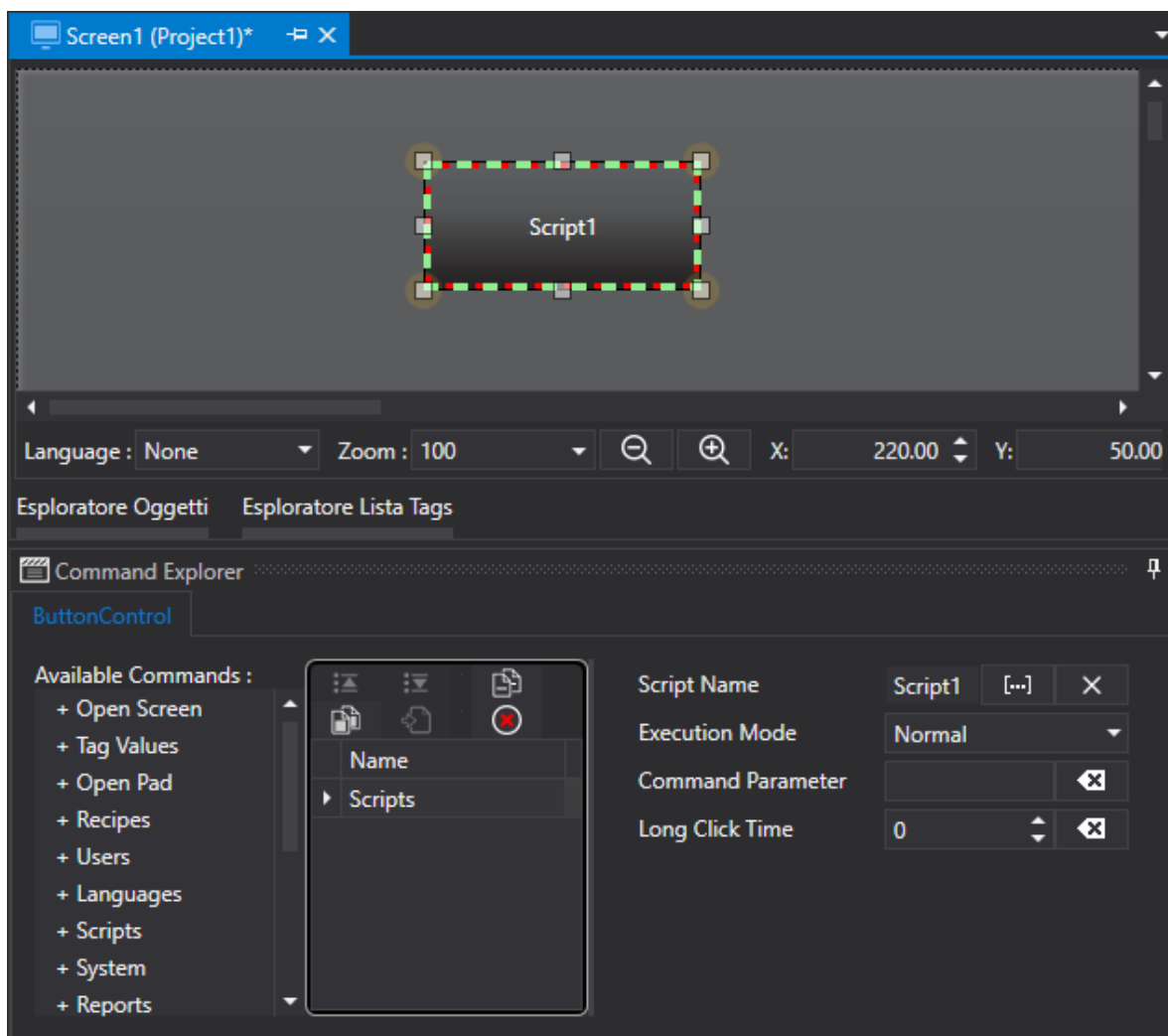
A Script Resource can be run:

- with a command associated to a graphic object (Symbol or User Control)
- with a command associated to an event
- with a command associated to an alarm



As the commands are launched from the Movicon.NExT Client, the Scripts run on Command are those scripts that work on the same PC in which the Project's Client is running. This means that if a script is set to run on a command associated to an event, the script will be run on every computer where the Project's Client is running.

For example, if you wish to associate the execution of a Script Resource to a button, simple select the button and open the Command Explorer Window, select the Script item, then the Script resource to run, as shown in the screen shot below.



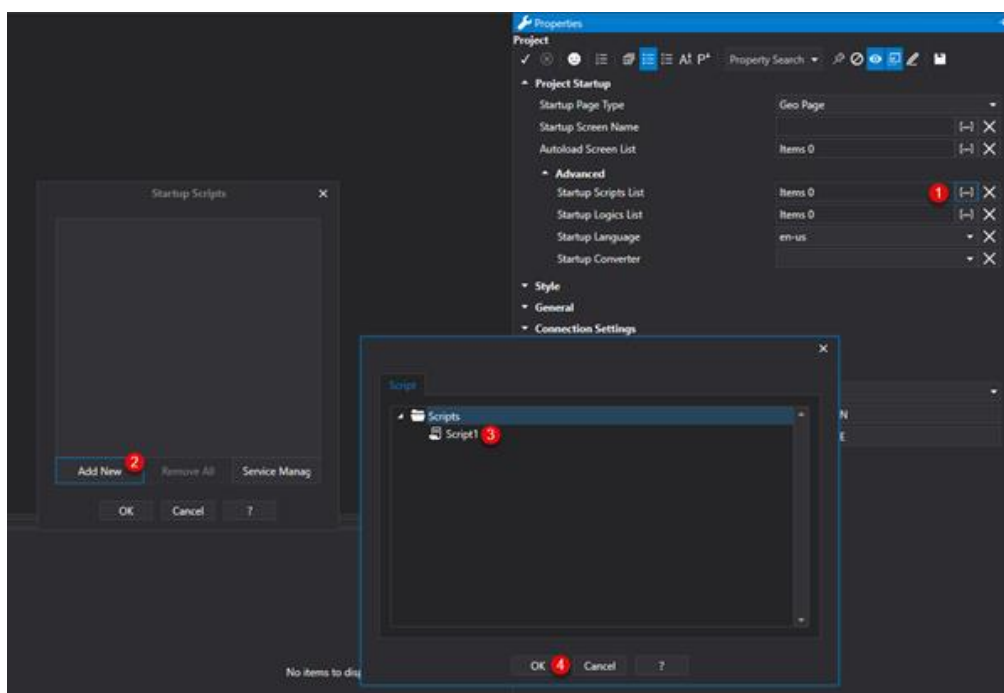
For further information on the Command properties please see the topic on: "Comando Esecuzione Script" , chapter: "Development Environment > Command Explorer Window"



The stop command stops the running Script Resource and unloads it from memory. In cases where more Script Resources are running in the same thread, the stop command will unload the Basic Script it relates to while the other Scripts running in the same thread will only be stopped.

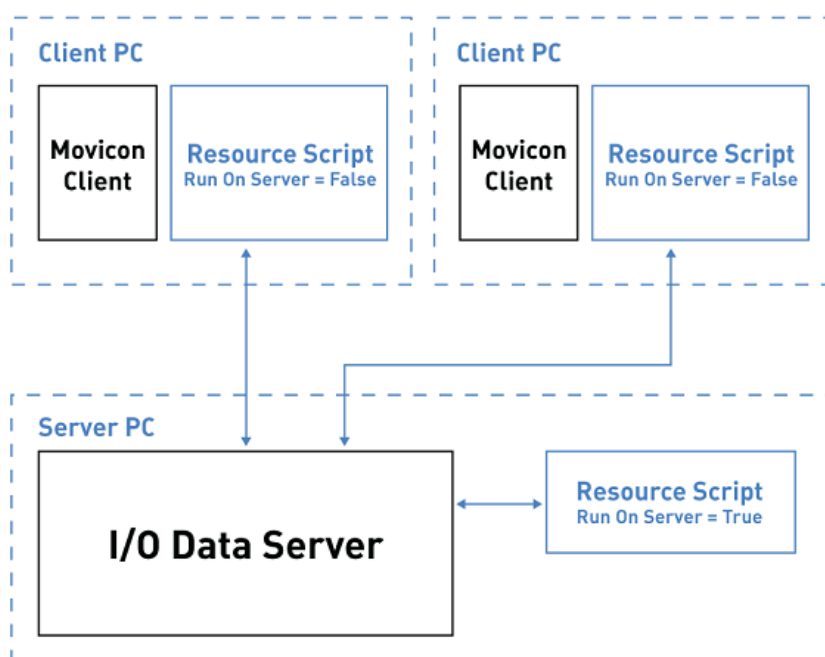
## Run at Runtime Startup

A Script Resource can be automatically run at the project's runtime startup. To use this functionality, you will need to access the appropriate "Startup Script List" settings from the project's properties by selecting the Script Resource from those available.



Once you have selected Script Resource to run at startup, in addition to the other startup options "Normal", "Synchro" and "Shared" (please see topic on "Run Script Command" for further details) you can also select the "Run on Server" option.

The "Run on Server" option allows you to choose whether the Script should run with every project client or started up on the machine where the projects I/O Data Server is running as shown in the diagram below:



In cases where you choose the "Run on Server" option, it will be possible to run scripts within the Script Service. This option is necessary in cases where the I/O Data Server is run at the machine's startup as Windows Service. For further information on running processes as Windows Services please refer to the chapter on "Services Control Panel" in the topic on "Development Environment".



## Script Resource Properties

Each Script Resource has properties that are used to manage how it functions, its debug and control. To do this, simply select the desired Script and then access its settings through the Movicon.NExT "Properties Window".

### General

#### Cycle Time Tag

This field is used to associate a Tag in which the Basic Script resource's cycle time will be reported. This value is set in milliseconds and will update when the script has terminated from being run. However if there is an infinite loop within the Script, the cycle time value will remain set at zero.



For performance reasons, the Script starts running without waiting until the Cycle Time Tag has been subscribed to the Server. This may mean that the Cycle Time Tag may start updating after the Script has gone into execution.

#### Current Tag Status

This field is used to associate a Tag in which the Script execution state will be reported. The Script states are managed in bits whereby each Tag bit has the following meaning:

Bit 0: not used.

Bit 1: Error. The bit changes to True each time the Script goes into error.

Bit 2: not used.

Bit 3: not used.

Bit 4: DoEventing. The bit changes to True each time a DoEvent instruction is run within the Script.

Bit 5: Starting. The bit changes to True when the Script is in startup mode.

Bit 6: Stopping. The bit changes to True when the Script is in stop mode.

Bit 7: Running. The bit changes to True when Script is running.



For performance reasons, the Script starts running without waiting until the Cycle Time Tag has been subscribed to the Server. This may mean that the Cycle Time Tag may start updating after the Script has gone into execution.

#### Thread Priority

A basic resource can be executed with different priorities: High, normal and low:

**Lowest:** very low priority

**BelowNormal:** Less than normal priority

**Normal:** Normal priority

**AboveNormal:** higher than normal priority

**Highest:** maximum priority

#### Force Writing On Server

The property is set to false for default. When set to true, the script will force the writing of the value of a valuable unconditionally to that of the previous one. This will permit same value to be rewritten within a tag.

### Execution

### Stop Command Timeout

The timeout period for the stop command.

### Sleep Time

The sleep time is set in milliseconds and is used to lighten the processor's basic script execution workload. The higher the sleep time, the less the basic script will engage the processor to execute it. As a consequence the script will take longer to execute.

## Connection Settings

The Connection properties allow you to define connection sessions, with the relevant parameters, between Client and Server for Tags used by the Screen resources. These Session parameters will then be used instead of the default Session parameters opened by the Client towards the Server.

### Session Name

This property is used to indicate a private Server-Client connection within which to manage the Tags used by the Screen resources. If a Session is not specified in this field, the Recipe will use the entire project's default Session.

### Refresh Time

This parameter is only managed when the Screen resource's 'Session Name' property has been set. This value, which the Client passes to the Server when the Session is opened, determines the time with which the Server sends Tag update notifications to the Client.

### In Use Tags Update Time

This parameter is only managed when the Screen resource's 'Session Name' property has been set. This value, which the Client passes to the Server when the Session is opened, determines the time with which the Server keeps the Tags put in use by Client updated. For example, this value is then also passed to an eventual Communication Driver and used as the minimum time to execute Tasks.

### Not in Use Tags Update Time

This parameter is only managed when the Screen resource's 'Session Name' property has been set. This value, which the Client passes to the Server when the Session is opened, determines the time with which the Server keeps Tags that are not in use updated but which are still subscribed by the Client. For example, this value is then also passed to eventual Communication Drivers and used as the minimum time to execute Tasks.

### Disable when not in use

This parameter is only managed when the Screen resource's 'Session Name' has been set. When this property is enabled, the 'Inactive Item' status is set when the Tag is not use in the Client.

### Number of Items to be Removed

This parameter is only managed when the Screen resource's 'Session Name' property has been set. This parameter is used to set the number of inactive OPC Items to be removed at each time interval defined in the 'Item Removal Delay' parameter.

### Item Removal Delay

This parameter is only managed when the Screen resource's 'Session Name' property has been set. This parameter is used to define the delay time that when expires all the inactive subscribed OPC UA Items will be removed. The number of subscribed items that will be removed when delay time expires is indicated in the 'Number of Items to be Removed' parameter. The delay time is expressed in seconds.

### Always Use Safe Connections

This parameter is only managed when the Screen resource's 'Session Name' property has been set. When this option is enabled, the Client will try to connect to the Server connections declared safe by the relevant certificates for both local and remote connections. In those cases when the Server does not expose safe connections, the next available connection will be used even though unsafe.

## Debug

### Enable Log

Enabling this selection box, an xml file will be created in the script's folder for the Basic Script routine during runtime. This file will contain printed messages belonging to the script such Debug.Print. This file will remain in use until the script stops running due to performance reasons.

This file will always be regenerated at startup.

### Enable SysLog

when enabling this property, the messages from the Debug.Print function will also be printed in the project's Historical Log as well.

### 9.5.3. Basic Script Tags

Script code can be defined within Tag In Movicon.NExT so that they can be run in the I/O Data Server and therefore independently from any Clients present.



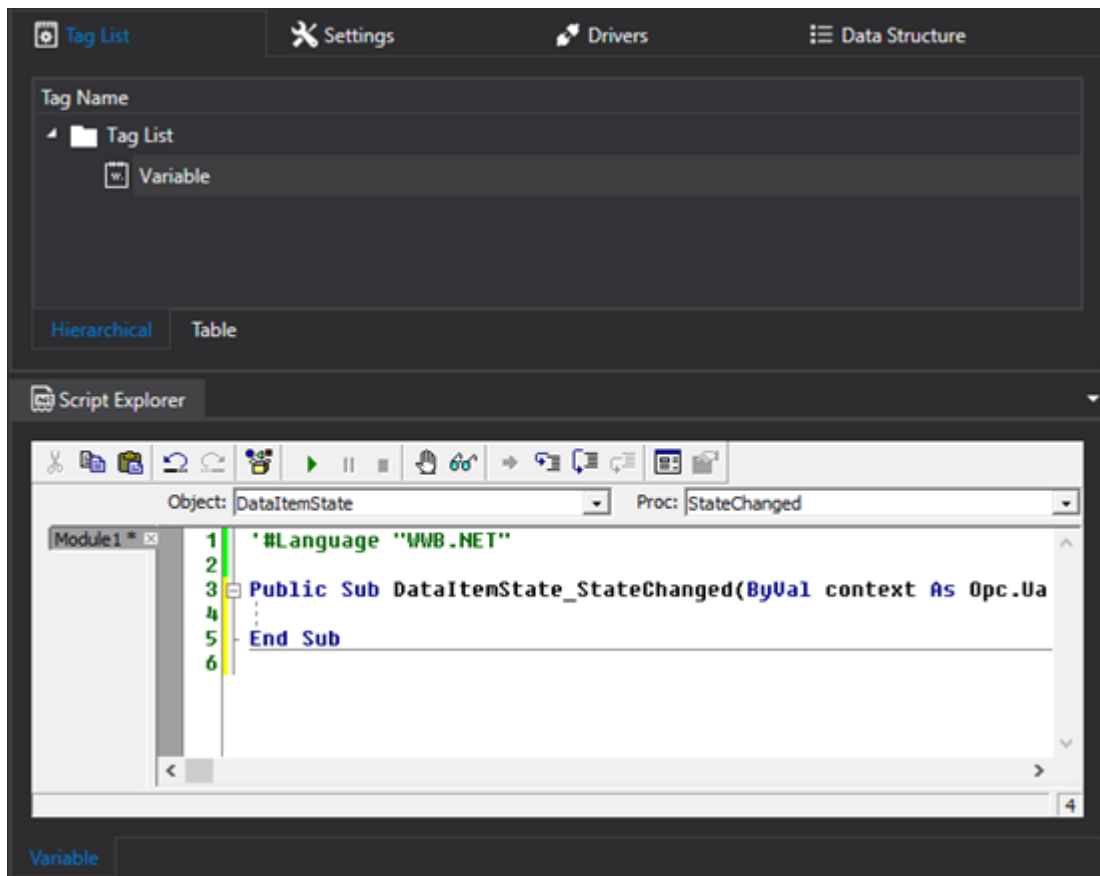
The scripts defined in Tags are all run in one thread for default.

Please take into account that when the project contains more than one Tag defined with one script, the scripts will not be run in parallel with each other.

When you need to run the Script of a Tag in a separate thread, you will need to enable the Tag's "Use Script in separate thread" property.

In order to define a Script within a Tag, you will need to open the I/O Data Server's Tag List and select the Tag within which to define the Script code and open the Script Explorer window. At this point, in order to associate a subroutine to an event, simply select the object from the "Object:" combobox menu and then the relative event from the "Proc." combobox menu which will automatically generate the subroutine within which it will be possible to edit the script code.

The screen shot below shows a script which has been defined in the StateChanged event of the "Variable" tag's DataItemState object.



The events to associate a Script in a Tag are:

#### **DataItemState/AnalogItemStateDocument > StateChanged**

The subroutine associated to this event executes when the value of the TimeStamp or Tag's Quality changes.



For the StateChanged event to function, the Tag must be in use! To force a Tag as always in used, simply set the Tag's "Always in Use" property to True.

#### **ServerScriptManager > Init**

The subroutine associated to this event is executed when the Tag is initialized, meaning at the startup of the I/O Data Server

#### **ServerScriptManager > DoEvents**

The subroutine associated to this event is executed around every 500ms.



The cycle time of 500ms is not deterministic.

#### **ServerScriptManager > InUseTag**

The subroutine associated to this event is executed when the Tag becomes "Active" therefore "In Use".

#### **ServerScriptManager > TerminateTag**

The subroutine associated to this event is executed when the Tag is removed when the I/O Data Server is turned off.



If you set a breakpoint in the Tag's code, the debug window will open at server startup even when the method has not yet been called.



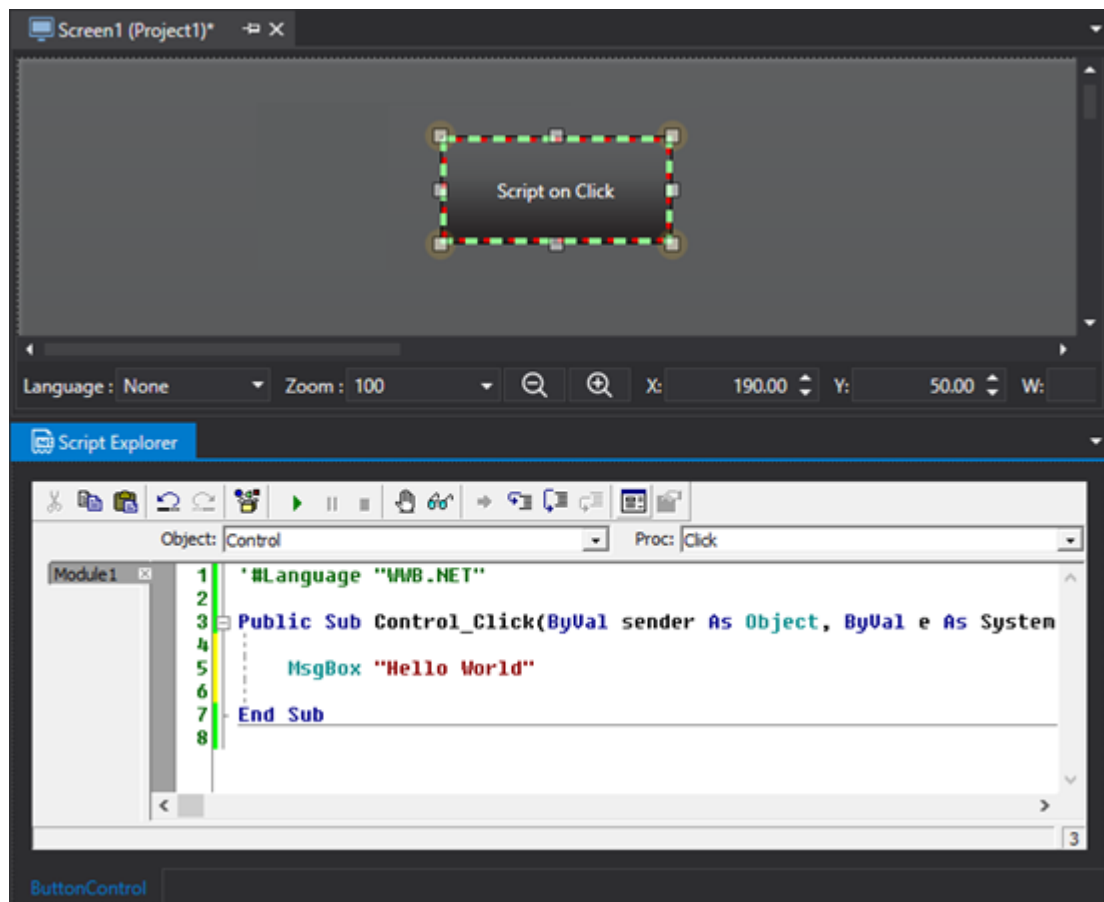
The length of the script code that can be inserted inside a Tag cannot not be more than 32767 characters.

### 9.5.4. Code in Basic Scripts and in Symbols

Scripts can be defined In Movicon.NExT and run in the project's Client by defining them in graphic objects (User Controls or Symbols) or in Screens. As a difference to the Script Resource, these Scripts do not have the "Sub Main" as an input point but allow subroutines to be defined and executed for each event generated by the object or screen.

#### Scripts in Objects

In order to define a Script within a User Control or Symbol, you will need to select it and open the Script Explorer window. To associated a subroutine to an event, simply select the object from the "Object:" combo box menu and then select the relative even from the "Proc." combo box menu. This will automatically generate the subroutine within which it will be possible to edit the script code. In the screen shot below a script has been defined in the Click event of a Control Button Object.



By referring to a Button type object, we can summarise the different Objects in this way:

- Object Control: exposes the events linked to the Button Control such as Click, Loaded, Unloaded and all the events linked to the Mouse and Touch management.
- Object Entity: the Entity (the control's parent) exposes the events linked to the Button's property variations such as the VariableChanged meaning the variation of the value or the quality of the Tag associated to the Button.
- Object Document: the Document (the Entity object's parent) is the Screen in which the Button is inserted. The events available are Activated, Deactivated and ScreenLoaded.
- Object Parent: the Parent (the Document's parent) is the reference to the object which called the Screen. The available Disposing event is called when the screen is loaded from memory.

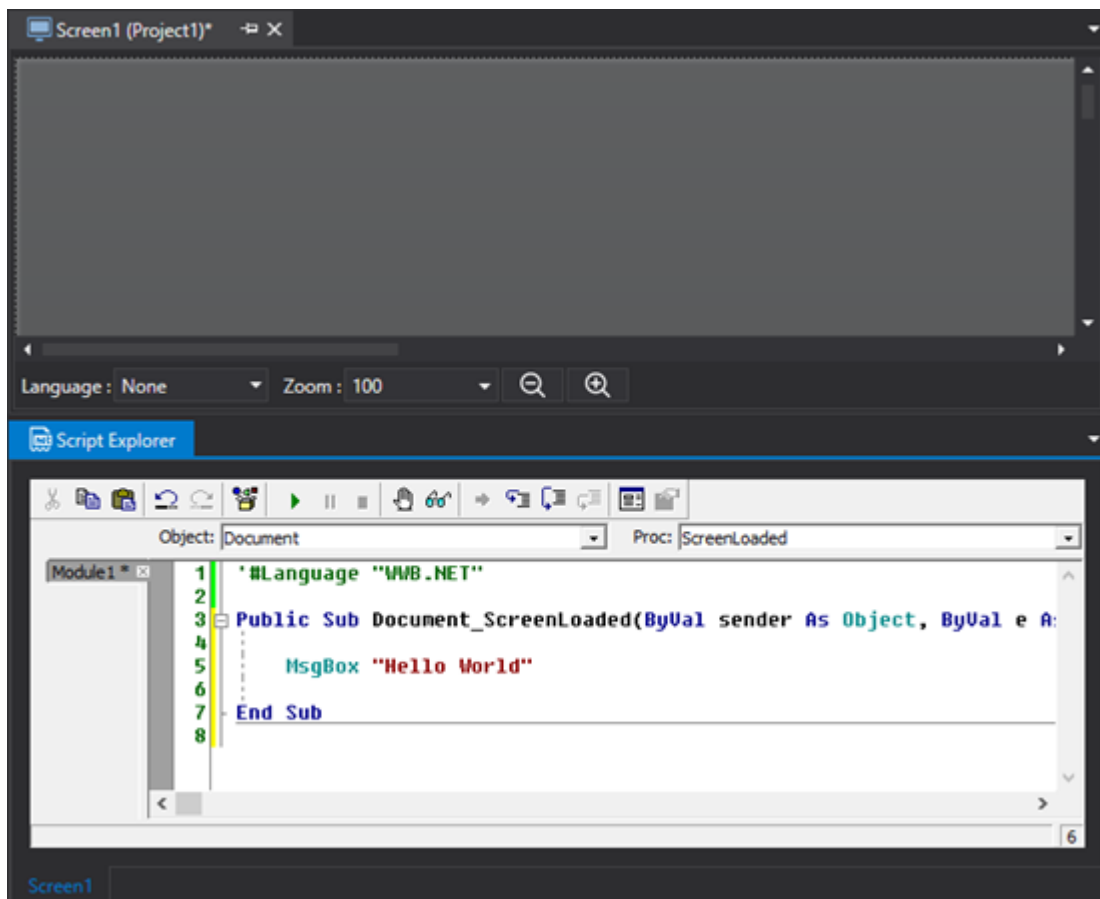


The code in Objects is not initialized straight away when opening the Screen that contains it but only in the presence of the "loaded" event or when the user starts interacting with the object.

### Scripts in Screens

In order to define a Script within a Screen, you will need to open the Screen, then click at the bottom of the screen to open the Script Explorer window. Once opened you can then associated a submarine to an even simply by selecting the object tfrom the "Object:" combo box menu and then the relative event from the "Proc." combobox menu to automatically generate the subroutine inside of which the script code can be edited.

The screen shot below shows a script defined in the ScreenLoaded event of the Screen1 Screen.



By referring to a Screen we can summarise the different Objects in this way:

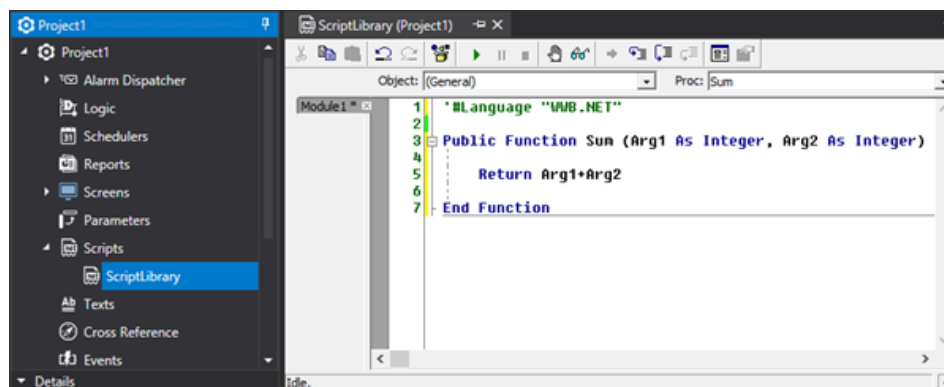
- Object Document: the Document is the Screen of which the Script is edited and in which the Activated, Deactivated e ScreenLoaded e ScreenUnloaded events are available.

- **Object Parent:** the Parent (the Document's parent) is the reference to the object which called the Screen and in which the Disposing event is available and which is called when the screen is loaded from memory.

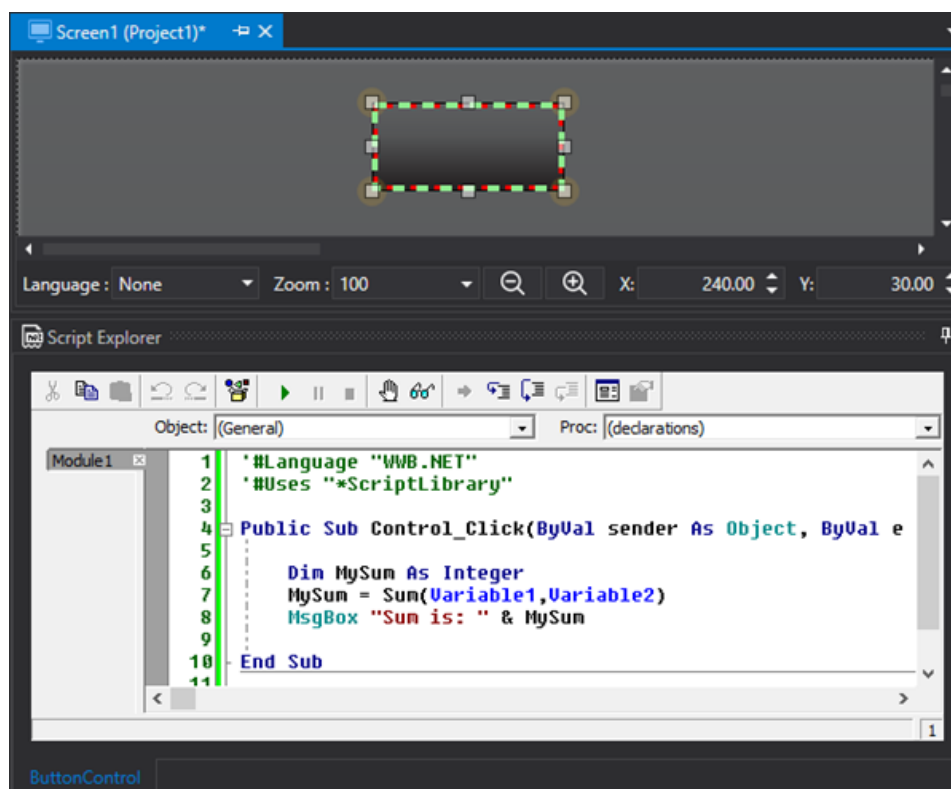
### 9.5.5. Basic Script Libraries

In cases where some Functions or Subroutines are frequently re-used in different points of the project it is possible to define a Script so that it acts as a collection or library of Functions.

The Script Library is in fact a Script Resource in which the "Sub Main" is omitted and where several functions and subroutines are added to be re-used in the project as shown in the screen shot below:



Inside the Script Resource or scripts of Objects/Screens it is possible to reference the functions within the Script Library by means of using the #Uses clause as shown in the screen shot below:



The #Uses clause syntax is:

```
'#Uses "*" <Script_Name>'
```

where <Script\_Name> is the name of the Script which acts as a library and which contains the function or script functions to be called.

The above example will become:

```
'#Uses "*"ScriptLibrary'
```

In cases where the Script Library is inserted inside a folder, the syntax of the #Uses clause will become:

```
'#Uses "..\" <Folder_Name>\" <Script_Name>'
```

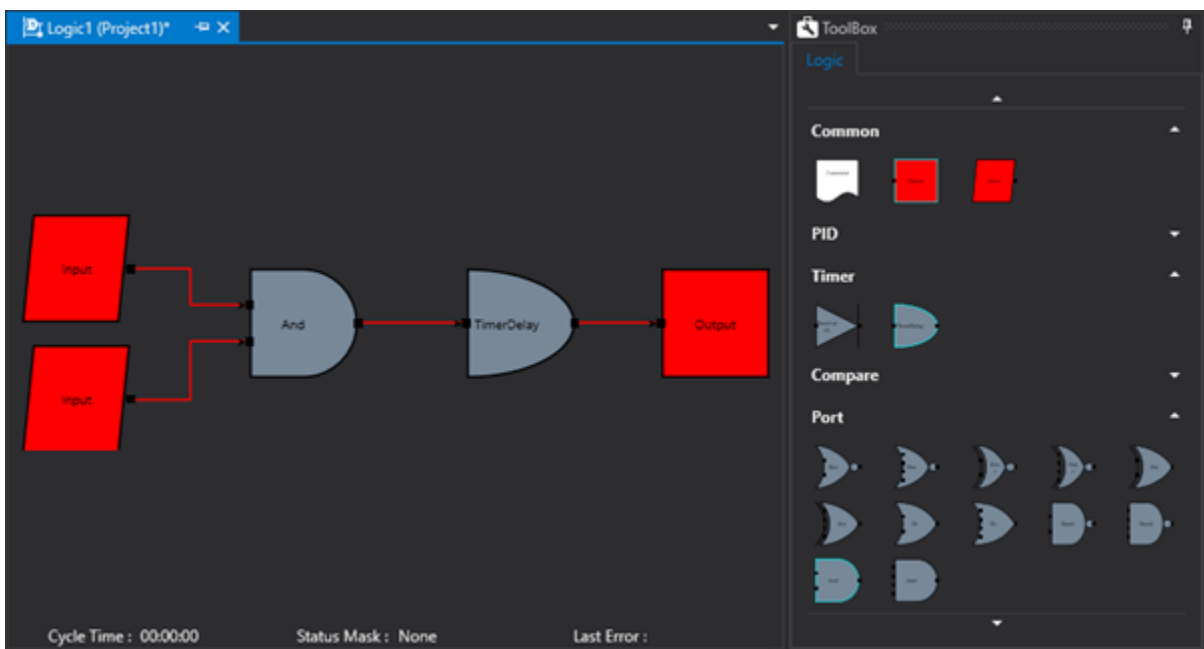
where ..\ indicated the root of the Script resource.



# 10. Logic

## 10.1. Introduction to using Logic

Sequential Logic Graphics is a sequential logic editor that can be programmed by inserting logic function block gates to which project Tags can be associated when consented. Logic editing is done graphically by inserting the logic function block gates in the editor and then connecting them to each other by using their input and output connectors. To create a connection simply click on an output connector of a function block gate (e.g. an "Input" function block gate, etc) and, by keeping the mouse key pressed down, drag the input connector with the mouse pointer to the gate desired (eg. a "Or" or "And" gate). The final result can be seen in the below image:



Sequential Logic function in the same way as that of a PLC's. This means that the Input variables are updated (e.g. those associated to the "Input" and "Compare" function block gates) at the beginning of a cycle run after which all the logic is processed. The Output variables are updated at the end of the cycle run (those associated to the "Output" function block gates).

## 10.2. Logic Function Blocks and Gates

### Logic ToolBox

The ToolBox activates when opening the Logic and it contains the function blocks that can be inserted in the logic. The ToolBox can also present the Tab with all the screen's objects but the screen's objects from the ToolBox or from the Symbol Gallery cannot be inserted inside logic. Likewise, blocks from the Logic's Toolbox cannot be inserted in a Screen.

The function blocks that can be used within logic are the following:

## Math Group

There are blocks available for "Add", "Subtract", "Multiply" and "Divide" operations that allow you to add, subtract, multiply and divide the input value with a preset value or with a value contained in a second tag.

## Port Group

The port logic available are: "RisingEdge", "FallingEdge", "AND", "OR", "NOT", "NOR", "NAND", "XOR", "NOR" and "XNOR". These ports can have two or four inputs. The output of these logic ports are activated when the input values satisfy the specific conditions of the different logic ports.

## Script Group

The two block with 2 or 4 digital inputs (Bool) that allow JavaScript code to be executed belong to this group.

This programming language (with full ECMAScript 5.1 support) adds to the VB.NET language that is already present in the platform, and is particularly handy to use in WebHMI Runtime, where the VB.NET scripts are not available.

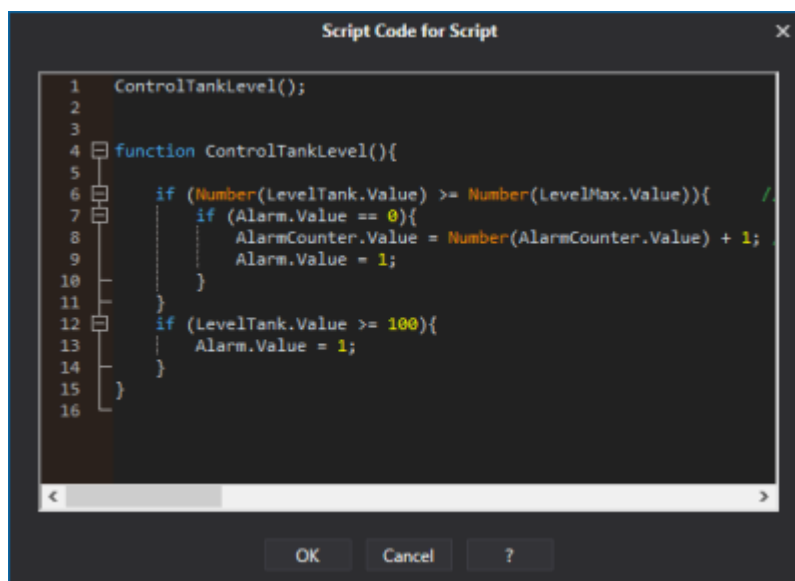
If you wish to know more about this programming language, please go to this website: Web:

<https://javascript.info>

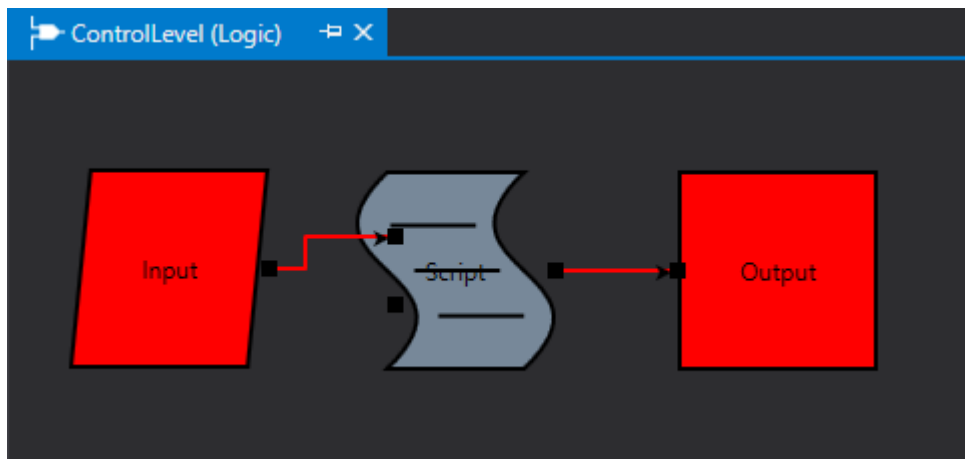
which provides a series of simple tutorials divided by argument.

By using the JavaScript instructions, you will be able to execute instructions in Tags associated to the object's Tag List within the blocks.

To read/write the values of Tags associated in the Tag List property, simply use the "<Variable\_name>.Value" syntax in the JavaScript code, as shown below:



Note that it is necessary to have at least one Input block and one Output block to allow the execution of the Javascript block as shown below:



Input and Output (which can only be digital) can be read and written from script code.  
To read the value of digital inputs:

```
MyVariable1 var = Input0
MyVariable2 var = Input1
MyVariable3 var = Input2
MyVariable4 var = Input3
```

where Input0, Input1, Input2 and Input3 are the first, second, third and fourth digital input respectively. Lastly, the code block can terminate with a "return true" or "return false" instruction to set the boolean value assigned to the output.

### General Group

The Input and Output ports, Comment blocks and "Set" blocks belong to this group.

The "Input" block allows input Tags to be associated for the Logic.

The "Output" block is useful for writing a defined numeric value in the associated Tag (or the value of the Tag associated to the "Tag Setting Value" property), when the input condition is True.

The "Comments" block is only needed to enter comments in the logic and it does not have any influence on running the logic.

The "Set" block sets the Tag associated with the defined numeric value (or the value of the Tag associated to the "Tag Setting Value" property), when the input logic condition is True.

### PID Group

The PID block (Proportional-Integral-Derivative) is a negative feedback system largely used in control systems. Thanks to an input that determines the actual value, it can react to any positive and negative errors tending towards the value 0. The reaction to the error can be regulated and this makes this block very versatile.

### Timer Group

There are two function blocks available in the Timer Group called "TimerDelay" and "TimerLatch". The "TimerDelay" starts counting when the input condition is verified (True) and when the time expires, it activates its output. The lack of the input condition will reset the count and consequently reset the output to zero. The "TimerLatch" activates the output upon verifying the input condition (True) and the time expires it will reset its output to zero. The lack of the input condition will zero the count and as a consequence resets the output to zero.

### Compare Group

The "Compare" block performs a comparison between two Tags or between a Tag and a constant value:

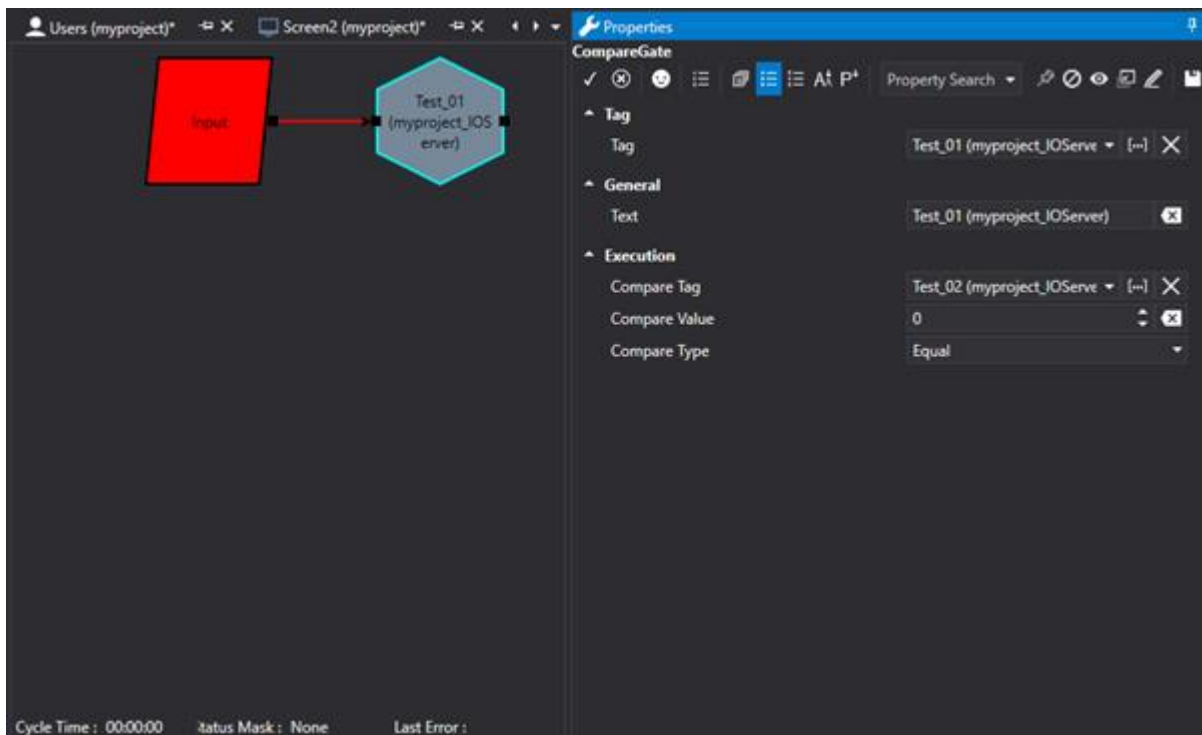
- When two Tags are associated to the block by means of the appropriate property, the comparison will be performed between the two tags.
- When only one Tag has been associated to the block, the comparison will automatically be performed in respect to the constant value defined in the appropriate property.

The comparison can also be performed with String type Tags. In this case, however, the comparison type performed can only be "Equal", while the comparison type performed for numeric values can be "Equal", "Major", "Minor", "Major or Equal", "Minor or Equal".

The comparison block will activate its own output when the input condition is verified (True) and the result of the comparison is verified.

## 10.3. Assigning TAGs

Assigning Tags to ports, that consent this operation, is done by using the same procedures used for assigning tags to objects on screen. Therefore, the tags can be selected from the Property Window or dragged directly from the Project window to the function block of interest. In cases where the Tags are dragged onto the object, the tag path will automatically be reported in the function block's 'Text' property.



When dragging tags to a "Compare" function block, only the first of the two object tags will be assigned (the second tag should be assigned using the "CompareTAG" property as shown in the above image).

## 10.4. Logic Resource Properties

Each Logic resource inserted in the 'Project Explorer' window can be associated with Properties that determine its mode of functioning when run. In order to do this simply open the Logic within the workspace and then modify its settings using the Movicon Properties Window.

### Connection Settings

The Connection properties allow you to define connection sessions, with the relevant parameters, between Client and Server for Tags used by the Screen resources. These Session parameters will then be used instead of the default Session parameters opened by the Client towards the Server.

#### Session Name

This property is used to indicate a private Server-Client connection within which to manage the Tags used by the Screen resources. If a Session is not specified in this field, the Recipe will use the entire project's default Session.

#### Refresh Time

This parameter is only managed when the Screen resource's 'Session Name' property has been set. This value, which the Client passes to the Server when the Session is opened, determines the time with which the Server sends Tag update notifications to the Client.

#### In Use Tags Update Time

This parameter is only managed when the Screen resource's 'Session Name' property has been set. This value, which the Client passes to the Server when the Session is opened, determines the time with which the Server keeps the Tags put in use by Client updated. For example, this value is then also passed to an eventual Communication Driver and used as the minimum time to execute Tasks.

#### Not in Use Tags Update Time

This parameter is only managed when the Screen resource's 'Session Name' property has been set. This value, which the Client passes to the Server when the Session is opened, determines the time with which the Server keeps Tags that are not in use updated but which are still subscribed by the Client. For example, this value is then also passed to eventual Communication Drivers and used as the minimum time to execute Tasks.

#### Disable when not in use

This parameter is only managed when the Screen resource's 'Session Name' has been set. When this property is enabled, the 'Inactive Item' status is set when the Tag is not use in the Client.

#### Number of Items to be Removed

This parameter is only managed when the Screen resource's 'Session Name' property has been set. This parameter is used to set the number of inactive OPC Items to be removed at each time interval defined in the 'Item Removal Delay' parameter.

#### Item Removal Delay

This parameter is only managed when the Screen resource's 'Session Name' property has been set. This parameter is used to define the delay time that when expires all the inactive subscribed OPC UA Items will be removed. The number of subscribed items that will be removed when delay time expires is indicated in the 'Number of Items to be Removed' parameter. The delay time is expressed in seconds.

#### Always Use Safe Connections

This parameter is only managed when the Screen resource's 'Session Name' property has been set. When this option is enabled, the Client will try to connect to the Server connections declared safe by the relevant certificates for both local and remote connections. In those cases when the Server does not expose safe connections, the next available connection will be used even though unsafe.

## Execution

### Cycle Clock

This parameter indicates the logic's minimum cycle time. If the execution of a logic's cycle terminates before that of the cycle clock, the next cycle will start execution only when the Cycle Clock time has ended as well. If the logic's cycle takes longer than the Cycle Clock's time to execute, the next cycle will start immediately after it has terminated. The value entered here is in milliseconds.



For performance reasons, the Logic does not wait until the 'Cycle Time' Tag Cycle has been subscribed to the Server in order to start. Therefore, the 'Time Cycle' tag will update after logic has started running.

### Cycle Time Tag

This field is used to associate a Tag which will report the time value of the cycle being executed. The value is expressed in milliseconds.

### Current Status Tag

This field is used to associate a Tag to report the Logic's execution status. The Logic statuses are managed in bit, whereby each individual Tag bit has a specific meaning:

Bit 0: Error. This bit turns True each time the logic goes into error.

Bit 1: Reading Tags. This bit turns True each time the Logic's Input variables are read at the beginning of a cycle.

Bit 2: Writing Tags. This bit turns true each time the Logic's Output variables are written at the end of a cycle.

Bit 3: Waiting For Good Tags. This bit turns True when a tag associated to the Logic does not have a valid value. For example if a String or Array tag type has been used.

Bit 4: Cycling. This bit turns True each time a new logic cycle begins.

Bit 5: Starting. This bit turns True when the Logic is in the starting phase

Bit 6: Stopping. This bit turns True when the Logic is in the stopping phase.

Bit 7: Running. This bit turns True when the logic is running.



For performance reasons, the Logic does not wait until the 'Cycle Time' Tag Cycle has been subscribed to the Server in order to start. Therefore, the 'Time Cycle' tag will update after logic has started running.



The Running bit remains True for the whole duration of the logic's running time, while the Cycling bit turns True at the impulse of a new cycle beginning.

### Thread Priority

A Logic resource can be run according to priority:

Lowest: the lowest priority

Below Normal: a below normal priority

Normal: a normal priority  
Above Normal: an above normal priority  
Highest: the highest priority

## 10.5. Logic Execution

### Running and Aborting Logic

A Logic resource can be run either by using a command (by button or event) or by using the Project's "Startup Logics List" containing a list of logic to be run at project startup. In both cases it is possible to run logic with the following modes:

- **Normal:** Logic is started and kept running in a continuous cycle. The logic run can be stopped by using its 'Stop' command otherwise it will be stopped when the project stops running.
- **Synchro:** The Logic is run once only then it is stopped automatically. Therefore only one complete Logic cycle will be run.
- **Shared:** The logic is run in "debug" mode, meaning that it will be run in 'Normal' mode with the debug window opened for displaying the logic function blocks edited in design mode. The function blocks will be animated within the debug window according to the Tag values and resulting logic combinations. Closing the debug window will also stop the logic from running.
- **Stop:** The logic set in this command will be stopped.

### Run Logic as Service

When logic is inserted in the "Startup list", it will also become possible to select the option to start logic as Windows Service. In this way the selected logic will start when the PC is started up as Windows Server and therefore processed independently from the Movicon Client startup command. However, the Logic Service startup will still depend on the PlatformNExTIOserver service process that needs to be installed as Windows Service to enable this. For further information about this please refer to the chapter on "Start as Service".

If the service is not installed after the "Start As Service" option has been checked, the logic will start up with an external "LogicService.exe" process in respect to that of "Movicon.NExT" process. In addition, when the "Start As Service" option has been checked, it will not be possible to start logic with the "Shared" mode.

### Logic Simulation

There are two commands in the Project > Logic > Server Menu or toolbar that are used to run an initial simulation run of the inserted logic.

- The **"Run Simulation"** starts logic run. In this phase it will be possible to force the input block values by double clicking on the block's borders in order to verify whether the logic is flowing correctly.
- The **"Stop Simulation"** terminates the simulation. Logic editing mode will remain active even when the simulation has been run.





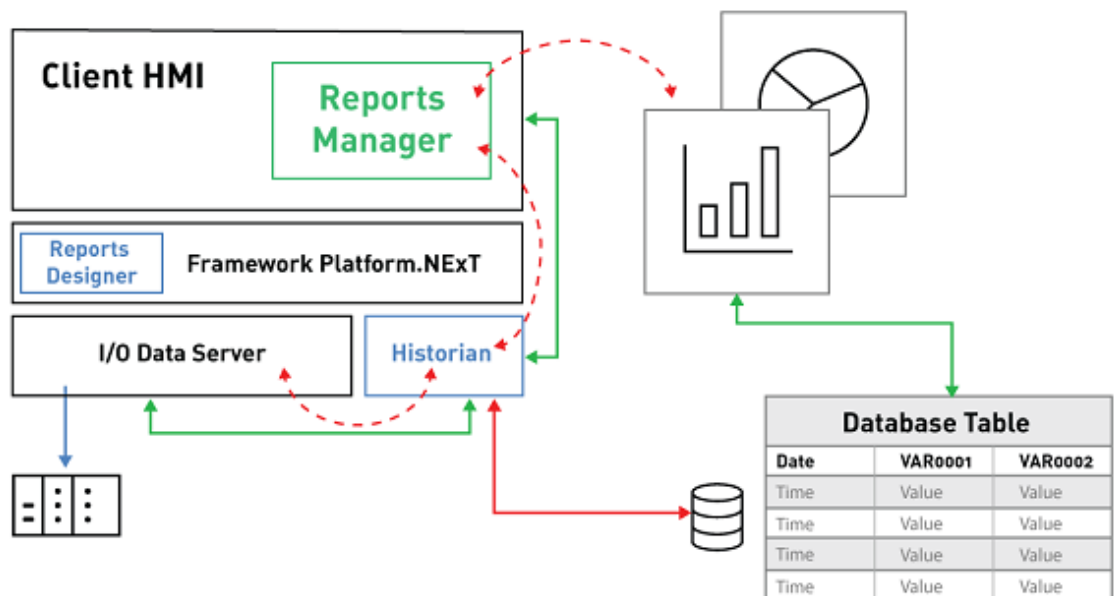
# 11. Reports

## 11.1. Introduction to using the Report Designer

The Report Designer is a powerful tool which is integrated in the Movicon.NExT Report Manager. It is used for creating editing and creating reports based on data recorded in the databases. These databases are the Reports' data sources and they are usually created by the project using the Historian and Data Logger functions. However, reports do not necessary have to use a database created and managed by Movicon.NExT as a data source.

The Report configuration interface is available from the platform's design mode environment and is located in the **Reports** resource listed in the project tree structure.

The Report Designer can be called by using a new report command in the "Reports" resource. This button will open the Report Designer enabling you to edit your report:



*This is an example of the Movicon.NExT workspace with the Report Designer.*

This command will open the Report Designer window. The Report Designer opens showing its "Tool Box" on the left and the "Report Explorer", "Field List" and "Grid Property" on the right. The report Layout is in the centre.

By using the Movicon commands the report will open with the database connection already configured with the original object's data table fields listed in the "Field List" for reference (DataLogger/Recipe table variable TraceDB or Historical Log).

The report is composed of "Band" objects, which are inserted using the "Insert Band" command accessible with a right click on the report layout area. The Bands are:

- **Top Margin:** one per page only. Represents the top limit of the report page
- **Report Header:** one per report only. The first thing printed and only on the first report page
- **Page Header:** page header and is printed at the top of every page
- **Group Header:** you can use more than one group header per report to represent the beginning of a data group

- **Detail: contains** the data to be displayed
- **Group Footer:** closes a data group and always corresponds to a Group Header
- **Report Footer:** printed at the end of the report on the last page
- **Page Footer:** closes each report page
- **Bottom Margin:** one only per page. Represents the report page's bottom limit

Only the Detail Band is needed to build a simple report. The fields you wish to display can be dragged from the "Field List" and when the report is executed these fields will show with all the records existing in the table.

### Filters

You can use the xtraReport object's Grid "FilterString" property (select the extraReport in the Report Explorer Window). This property must contain the expression criteria (WHERE (Es. [col] > 0, where "col" is the name of a data table column).

### Groups

You will need to insert a Group Header Band and use the "GroupFields" property for setting the second field to group data. You can also specify the data order in the same property.

### Data Summary

The report can perform summaries (totals, counts, averages,...) on data groups. This is done by inserting a "Label" band, linked to a field from the Field List. The "Label's" "Summary" property allows you to set the summary type you wish to create using the dialog window which contains:

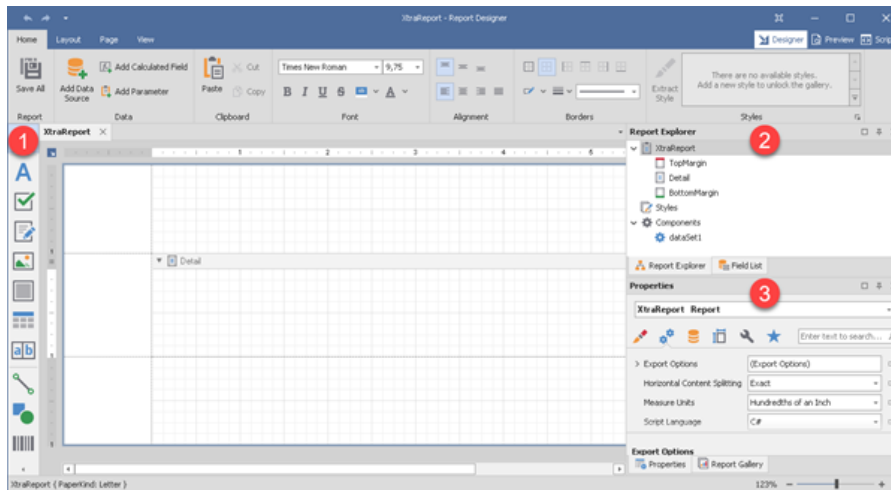
- **Bound field:** name of field in which the operation is to be performed
- **Summary function:** operation type to be performed
- **Format string:** format type for results where you can add measure units as text
- **Ignore NULL values:** specifies whether or not to ignore NULL field value records
- **Summary Running:** defines the environment in which to perform the operation. Obtains these values:
  - None = no calculation
  - Group = performed for all group members. This value resets at the beginning of the next group
  - Page = performed for the values presented on one page
  - Report = performed for all the values in the report

## 11.2. Report Designer

The Movicon.NExT Report Designer is the Reports management module that is used to edit and perform powerful analytical reports at runtime on data recorded by the Historian modules and available on database.

The Report tool is used for displaying and printing recorded process data according to a specific time and date range. Thanks to Movicon.NExT, managing Reports becomes easy ensuring the openness needed to obtain previews and prints of data analysis Reports adapted exactly to each individual need of the end user.

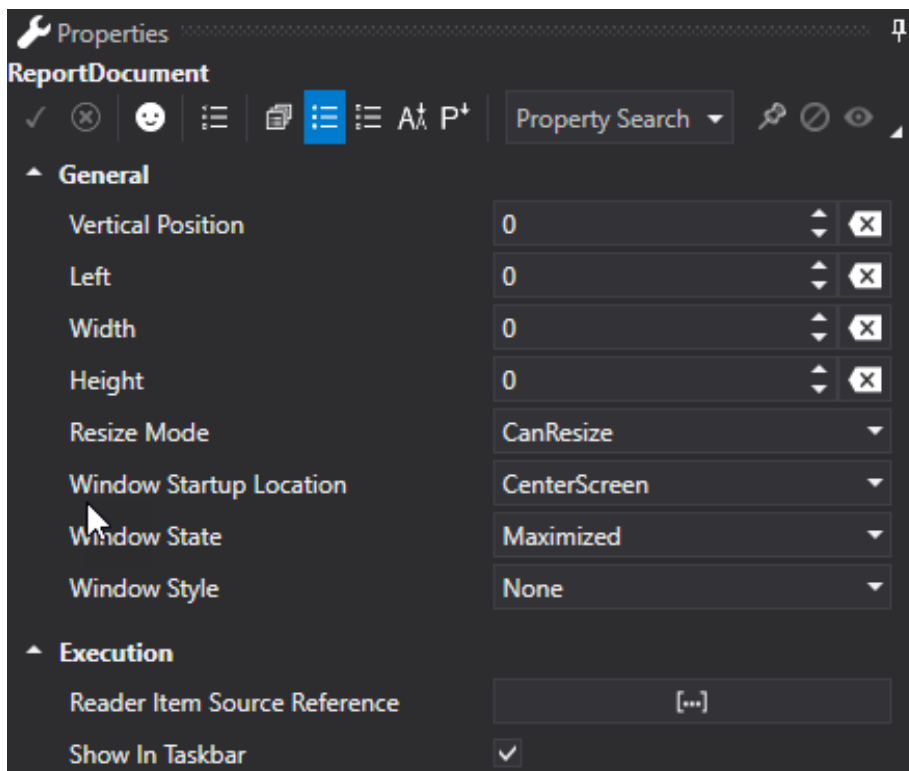
In addition to accessing historical data recorded by the Historian or Data Loggers, the Reports can also be used to access data by any period for any analysis or statistical analysis type. This also includes displaying calculations of totals, averages and differences by selecting time range type, production batch, operator and shift.



When you right click on the Report source and then on the new report item in the popup menu, or double click on an existing report, or right click on an existing report and then on the open command in the popup menu, the Report Designer window will open showing: (1) the "Tool Box" on the left and (2) "Report Explorer" and (3) "Properties" windows of the right. The report layout is shown in the center. Reports created in edit mode will be managed at runtime to represent analysis of historical data to which the "data sources" are connected.

## 11.3. Report properties

When selecting a Report from the Project Explorer Window and then clicking on the Properties tab, the Property window will display showing the main Report features that can be modified:



## General

### Full Path

This read only field indicates the complete path of the file in which the Report is saved.

### Vertical Position, Left, Width, Height

These fields are used for setting the position and size of the Report window when opened in runtime.

These values are only acknowledged when the "WindowStartupLocation" property has been set to Manual.

### Resize Mode

This property is used to select whether the report window can be resized or not. The possible values are:

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

### Window StartUp Location

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

- **Manual:** opens the Report according to the coordinates defined in the "Vertical Position", "Left", "Width" and "Height" properties
- **CenterScreen:** opens the Report in the centre of the screen
- **CenterOwner:** opens the Report in the centre of the calling screen

### Window State

This property is used to define the Report's open mode. The options are:

- **Normal:** the Report window will open with the default sizes defined in "Width" and "Height" properties
- **Minimize:** the Report window will open minimized i.e. reduced to an icon on the Windows bar
- **Maximize:** the Report window will open at full size

### Window Style

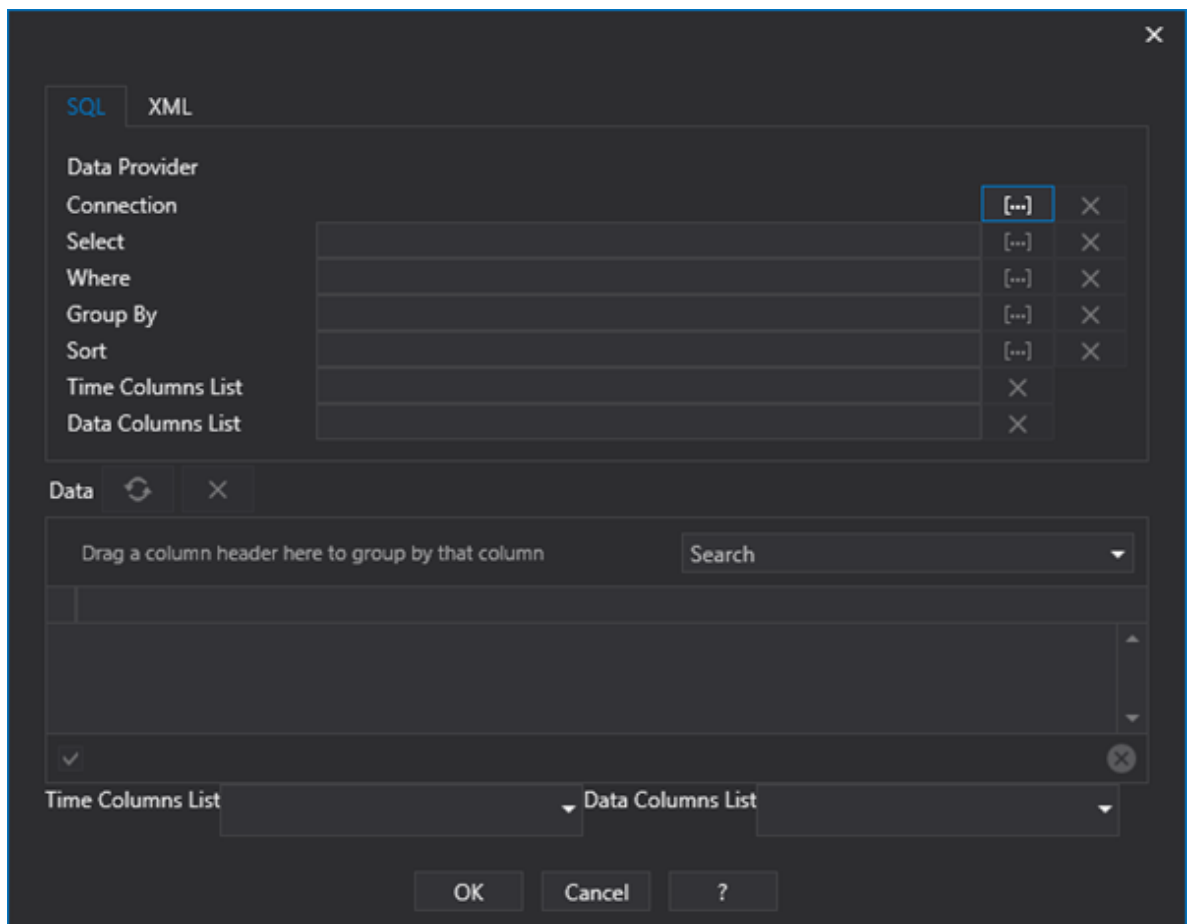
This property is used to define the Report's opening style. The options are:

- **None:** the Report will open in a window without a title bar
- **SingleBorderWindow:** the Report will open in a window with a title bar and a simple border
- **ThreeDBorderWindow:** the Report will be opened in a window with the title bar and a 3D border
- **ToolWindow:** the Report will open in a basic styled window with a title bar and a simple border

## Execution

### Reader Item Source Reference

This field is used to set the Report's connection towards the data source. In the Data Source Editor window, which opens when activating this command, you will find in addition to the selection of Servers and Data Bases that data selection queries with filter and order by clauses can be set as well.



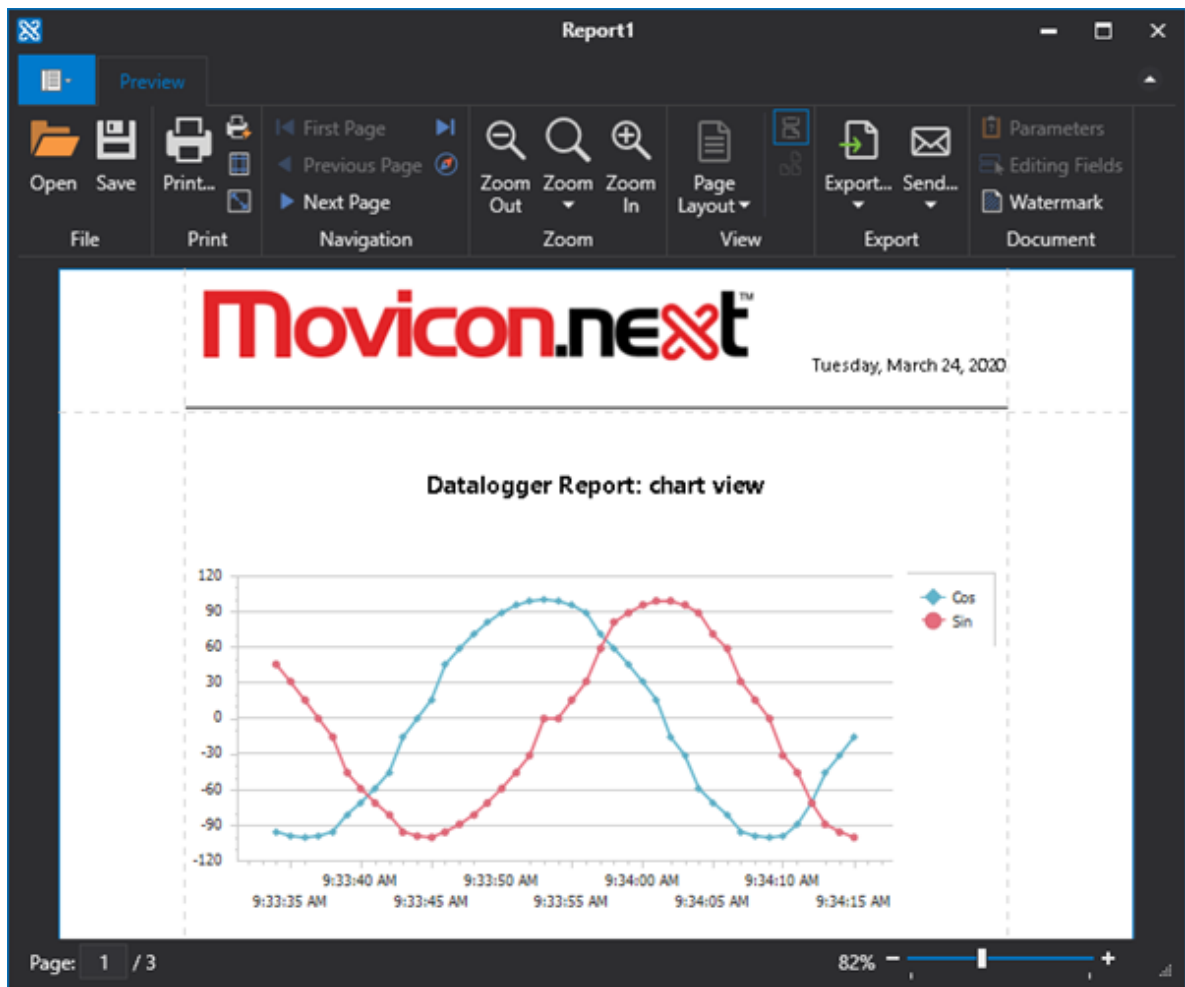
### Show In TaskBar

When this property is enabled the Report will also show in the Windows application bar when opened. Otherwise, it will not be shown When this property is disabled.

## 11.4. Report Execution

In runtime the report can be activated using a command assigned to a button for example (for further info see section on "Using Commands" in Movicon NExT) or by using a "ReportViewer" object (see "Toolbox objects", "Report Viewer" in Movicon NExT).

After activating the report at runtime (or in Preview mode from the design environment), you will be able to access the toolbox located at the top of the report as shown below:



All the labels within report can be localized in your preferred language. For further information please see the chapter on "Text Management"

## 11.5. Report via Web

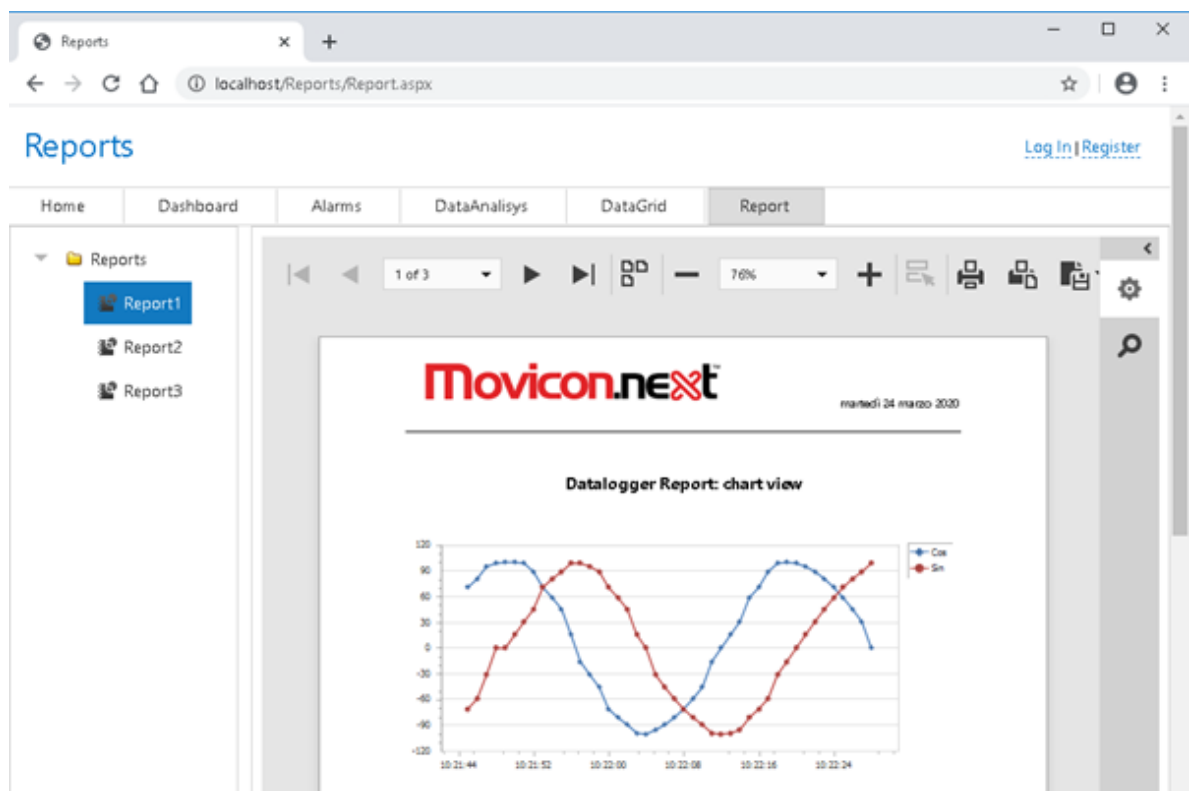
Displaying reports over the web is supported in both WebClient (WPF/HTML5) and WebClient (SVG/HTML5) as well as WebHMI.

### Reports in WebClient (WPF/HTML5)

After having deployed the project to use with WebClient (WPF/HTML5), the reports can be accessed by using the "Report" button from the "Header" toolbar.

If for any reason, the "Header" toolbar should not be visible, you can use the below url to access them:

`http://<indirizzo_server>/<nome_progetto>/Reports.aspx`



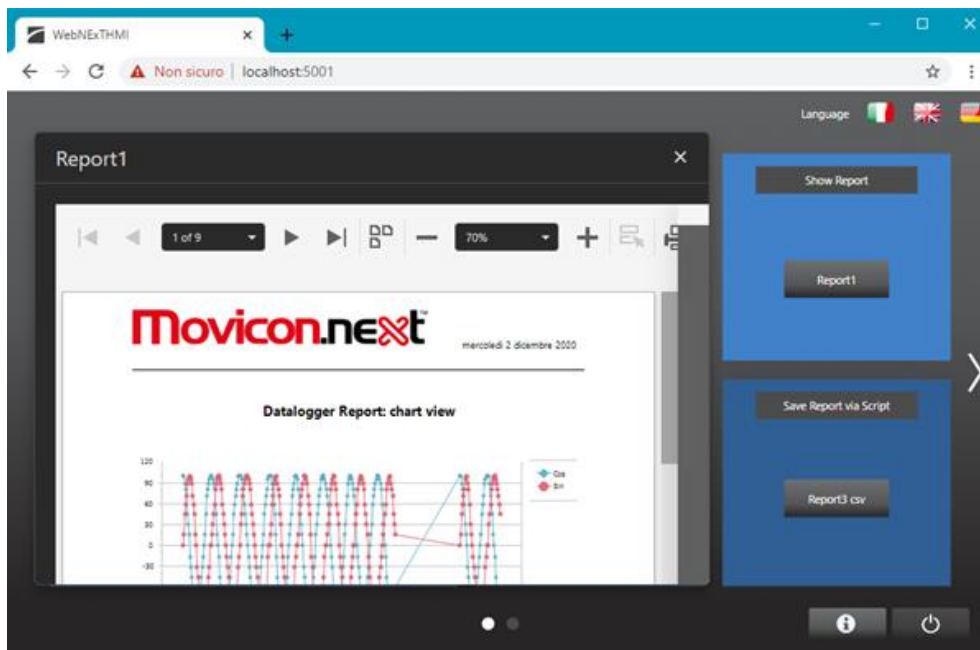
On the left you will find a list of available reports which will display in the center when selected. On the right a pane will show to insert additional report parameters if needed.



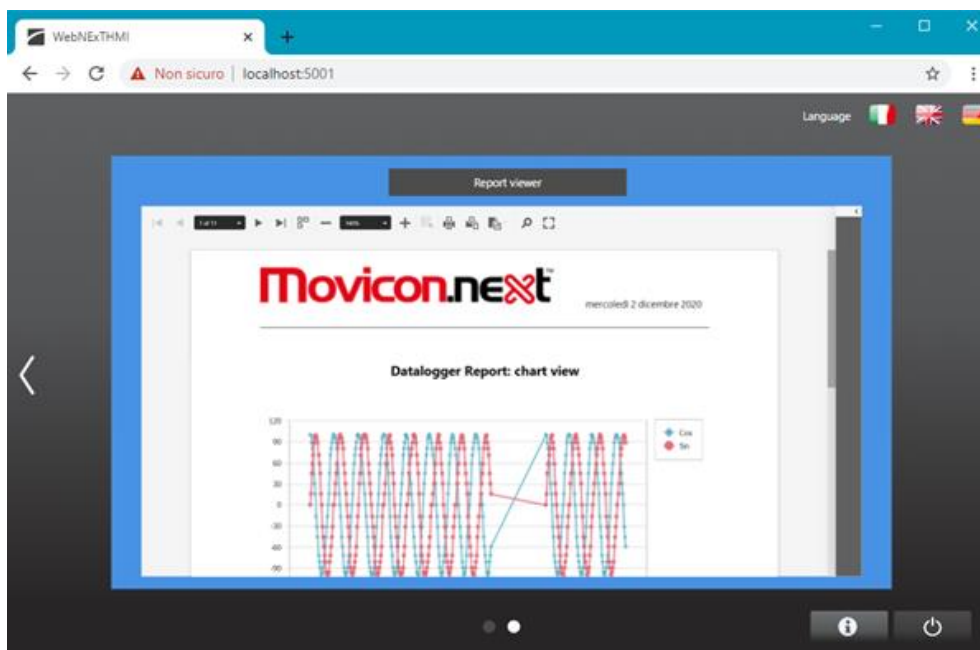
Please refer to the "Web Client" section for further details.

### Reports in WebClient (SVG/HTML5) and WebHMI

After having deployed the project to use with WebClient (SVG/HTML5) or as WebHMI, you will be able to access the reports by:  
using the associated "Report" commands such as the buttons in the screen shot below:



or by using the "Report Viewer" object:



When the WebHMI Runtime is running on the Linux platform, you will need to install the following libraries/packages on the target machine:

libgdplus, libicu-dev, libharfbuzz0b, libfontconfig1, libfreetype6, libpango-1.0-0, libpangocairo-1.0

For example, by using these commands:

```
sudo apt-get update
sudo apt-get install -y software-properties-common
sudo add-apt-repository 'deb http://deb.debian.org/debian sid main'
sudo apt-get update
sudo apt-get install libgdplus
sudo apt-get install libicu-dev
sudo apt-get install libharfbuzz0b
sudo apt-get install libfontconfig1
```



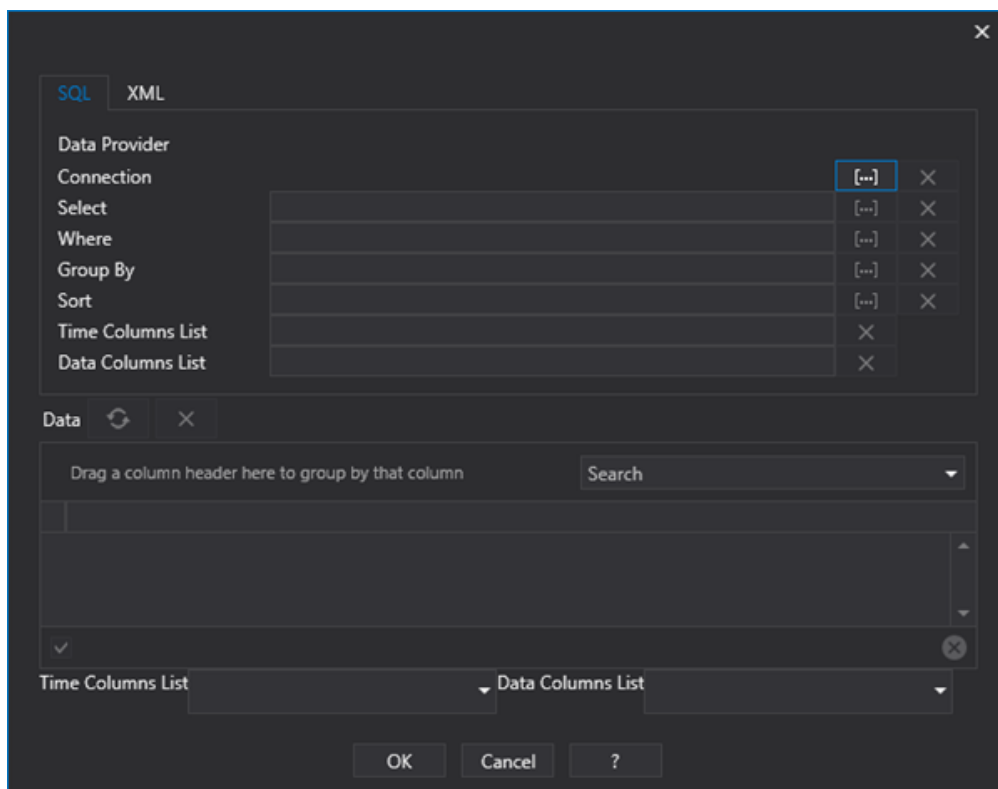
```
sudo apt-get install libfreetype6
sudo apt-get install libpango-1.0-0
sudo apt-get install libpangocairo-1.0
```

## 11.6. Passing parameters and DB Connections

The Report connection towards the database can be defined in different points and using different modalities as described below. The different modalities to pass parameters to the Report are also described towards the end of this topic.

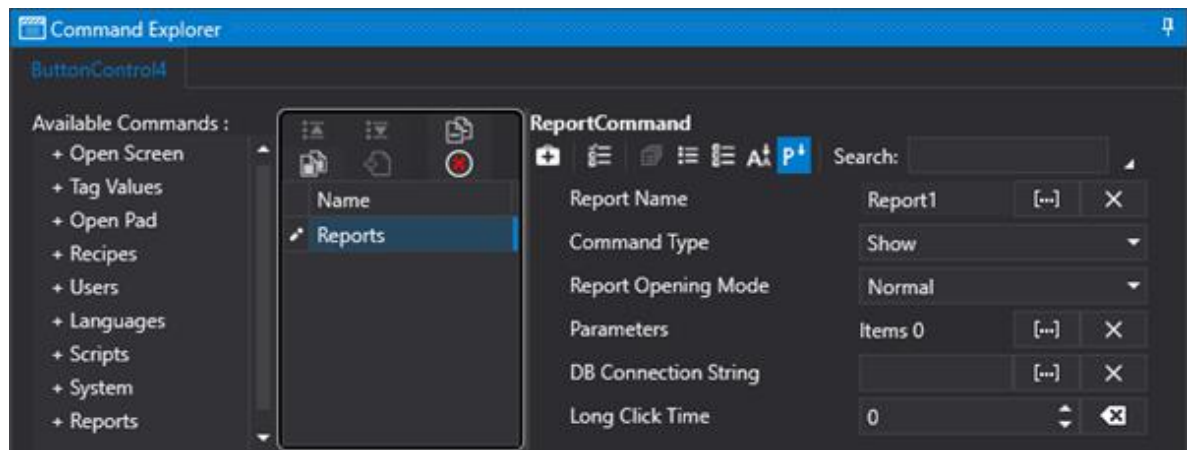
### Connecting to the Database by using the Report properties

There is a "Reader Item Source Reference" field In the Report Property Window that is used to set the Report's connection to a data source. In the Data Source Editor window that opens when executing this command, you can select the Server and the Database and also set a data selection query with filter and order by clauses.



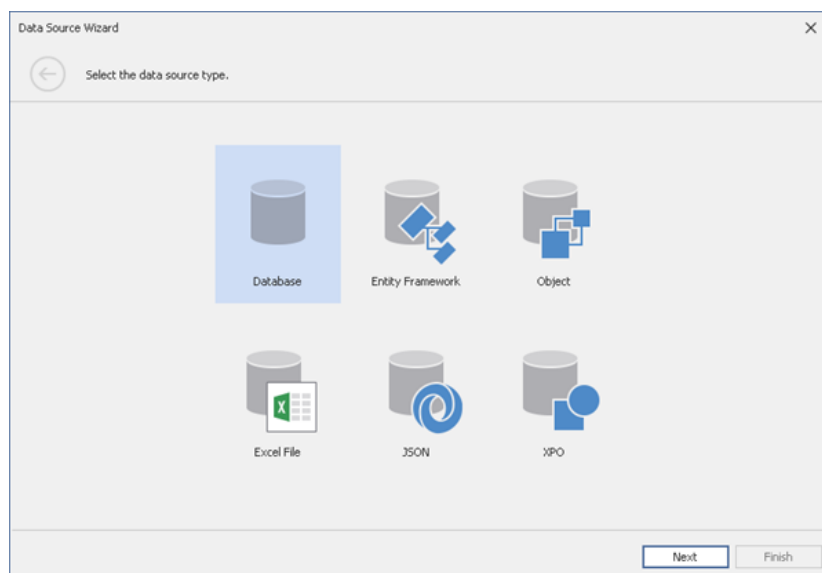
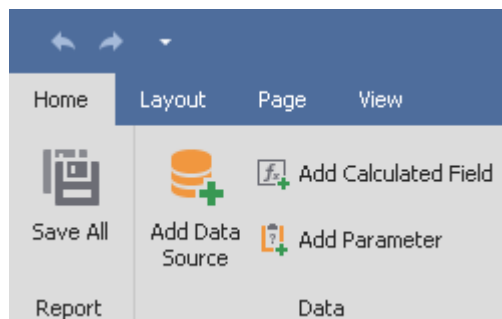
### Connecting to the Database by using the open command

The connection string that the Report uses to connect to the Database can be defined in the Report's Open command "DB Connection String" property as indicated in the following image:



### Connecting to the Database from within the Report

The connection string that the Report uses to connect to the Database can be defined directly within the Report with the Add Data Source command, as indicated below:



**Data Source Wizard** [X]

← Select the data provider and specify the connection properties.

Provider: Microsoft SQL Server  
 Server name: (local)\SQLEXPRESS  
 Authentication type: Windows authentication  
 User name:  
 Password:  
 Database: Reports\_IOServer

Next Finish

**Data Source Wizard** [X]

← Create a query or select a stored procedure.  
Data columns selected from specific tables and/or views will be automatically included into a separate query.

☒ Tables
 

- ☒ DataLogger
  - ☒ UtcTimeCol
  - ☒ LocalTimeCol
  - ☒ MillisecondsCol
  - ☒ UserCol
  - ☒ ReasonCol
  - ☒ sin
  - ☒ cos
- > ☐ UFUAAuditDataItem
- > ☐ UFUAAuditDataLog
- > ☐ UFUAAuditLogItem
- > ☐ XPOObjectType

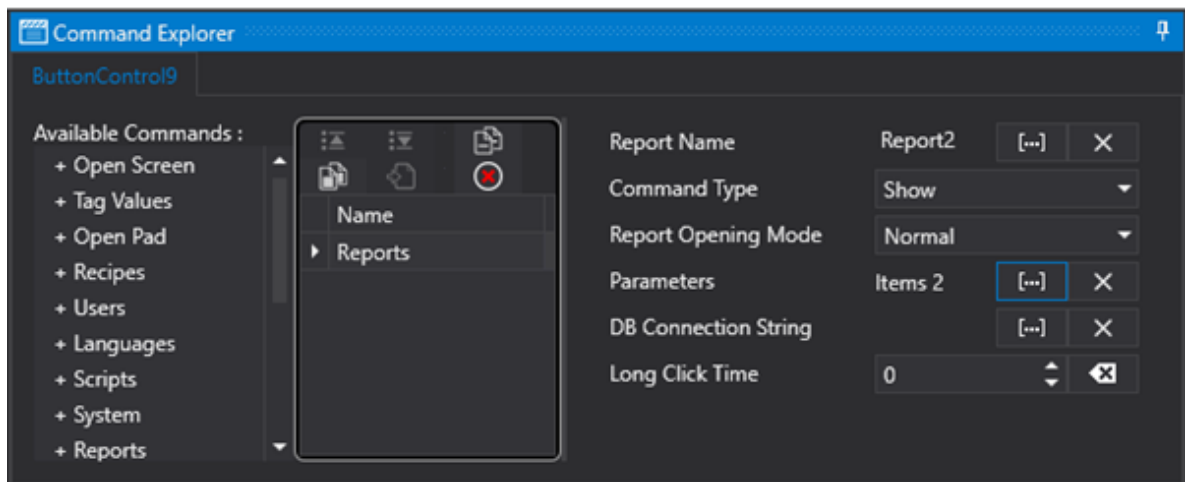
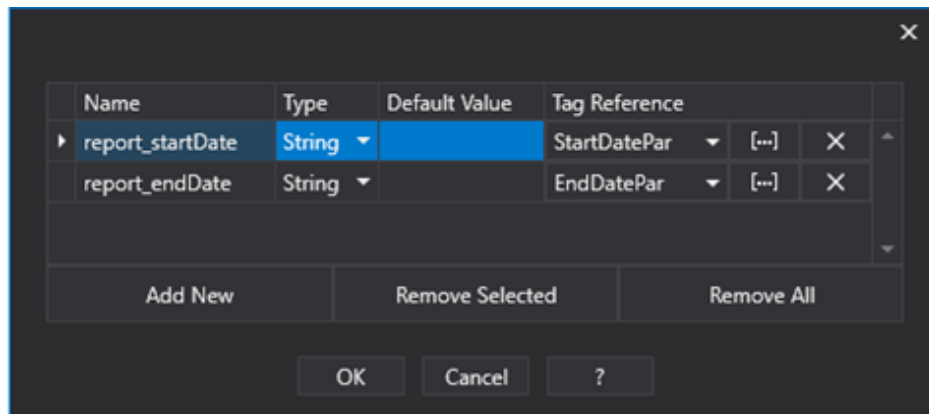
☐ Views

Manage Relations... Next Finish

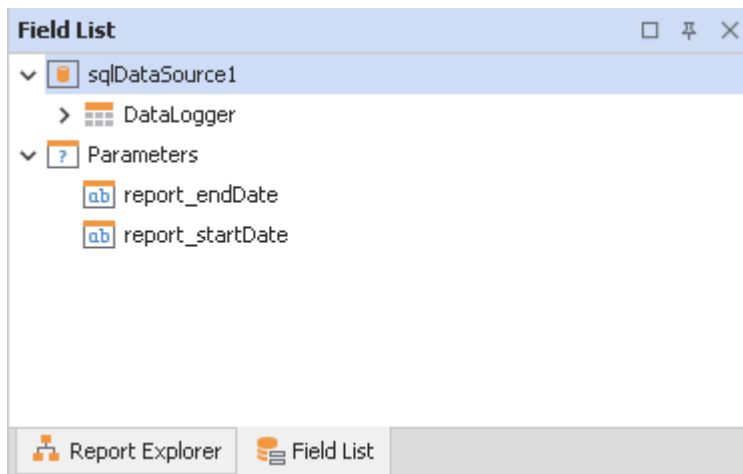
### Passing parameters to the Report

Parameters can be defined in the Report's open command which are useful for generating the Report, such as fixed values or as Tag values. The use of parameters is particularly useful when needing to generate a Report based on a query parameterized with the project's Tag values such as production start and end dates.

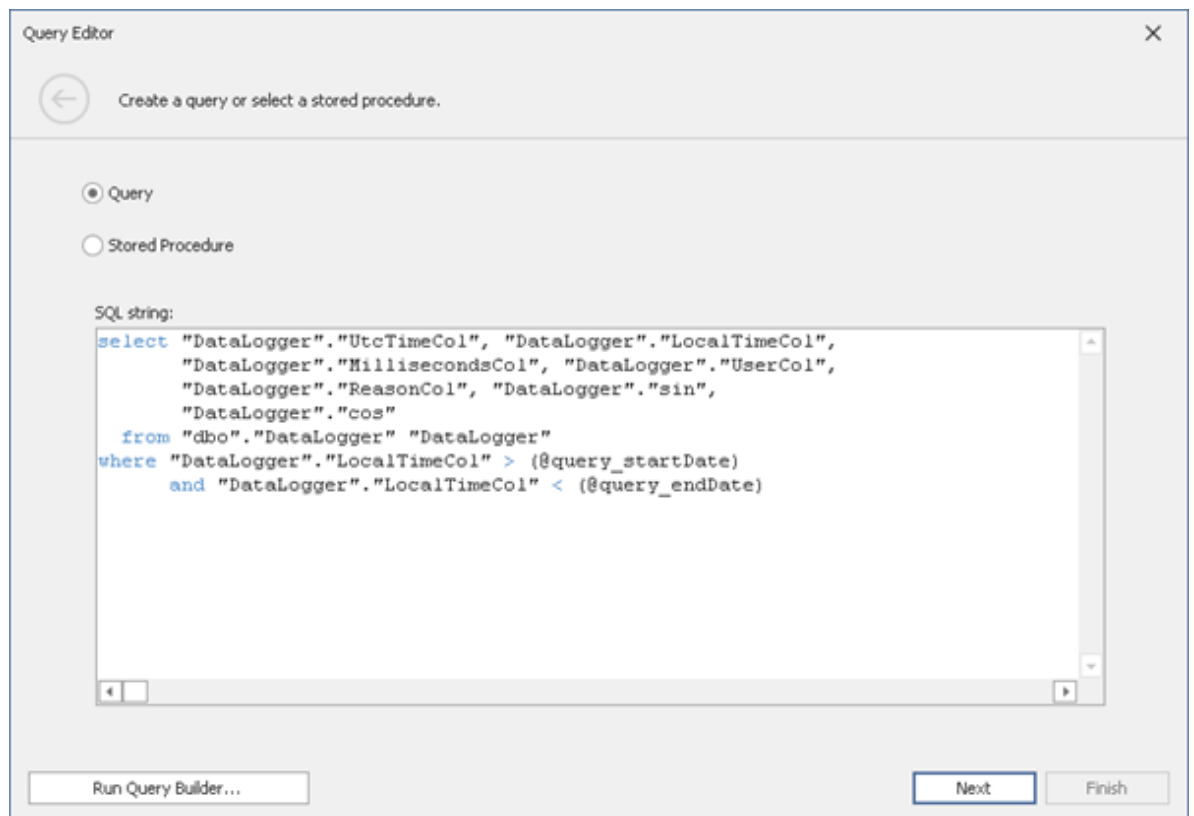
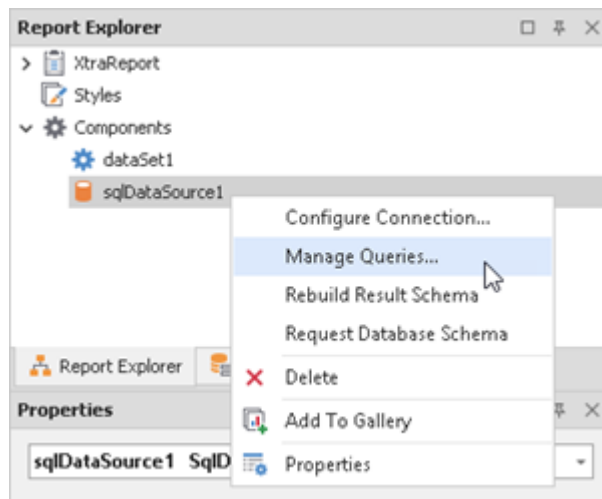
It is possible to specify parameters in the report's open command as shown below:



The parameter defined in the "Report\_startDate" and "Report\_endDate" command should correspond to the parameters defined within the report as shown below:



These parameters, defined in the Report's Field List, should correspond with the parameters defined within the Query as shown in the image below:



Query Editor

← Configure query parameters and preview the result.

Name	Type	Expression	Value
query_startDate	String	<input checked="" type="checkbox"/>	[Parameters.report_startDate]
query_endDate	String	<input checked="" type="checkbox"/>	[Parameters.report_endDate]

Preview... Add Remove Next Finish

## Special Parameters

Special parameters can be entered in the report to allow the automatic transition of information without needing to specify the information in the command relating to the report in Movicon.

The parameters are to created as string types and must correspond to the listed names below:

### UserName

This parameter functions only in WebClient and allows the name of the user who logged in and who is generating the report to be displayed.

### ConnectionString

This allows the report's default connection strings to be printed.

### CurrentCultureStrings

This allows transition of the information relating to the current language in the Movicon project so that the language change can be managed in the report.

# 12. Alarm Dispatcher

## 12.1. Introduction to using the Alarm Dispatcher

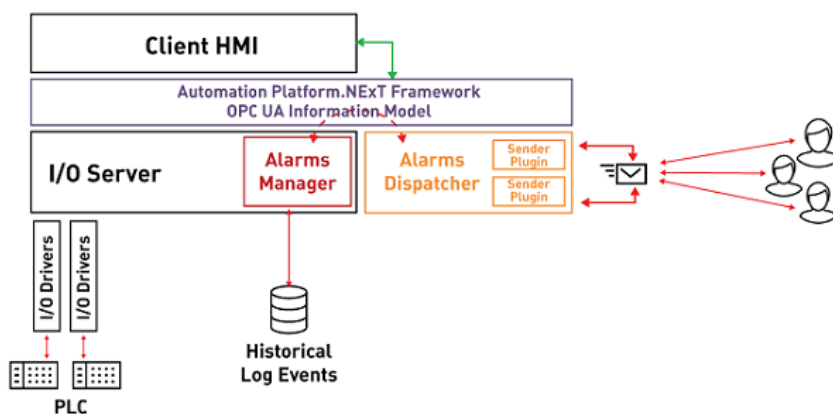
The Automation Movicon.NExT platform offers a function module for alarm and event notifications that can be set and sent to recipients such as on-call duty staff.

By means of using the integrated server module, the system allows you to define which events and alarms are to be notified (sent) and to which recipients by using the communication modes which are available as plugins.

The following plugins are currently available for notifying events:

- **SMS**
- **E-mails**
- **Voip**
- **Telegram**

A plugin for using vocal syntax is currently at work for future integration.



The Movicon.NExT Alarm Dispatcher is a module that operates as a data notification server by connecting to a Movicon.NExT project's I/O Data Server in order to access Tags in the most transparent way possible. The Alarm Dispatcher module is launched when the platform is started up in runtime. It can also be started up manually by using the "Start Server" command from the Alarm Dispatcher's Toolbar. The Alarm Dispatcher module technology is based on the OPC UA technology to ensure maximum interoperability and openness.

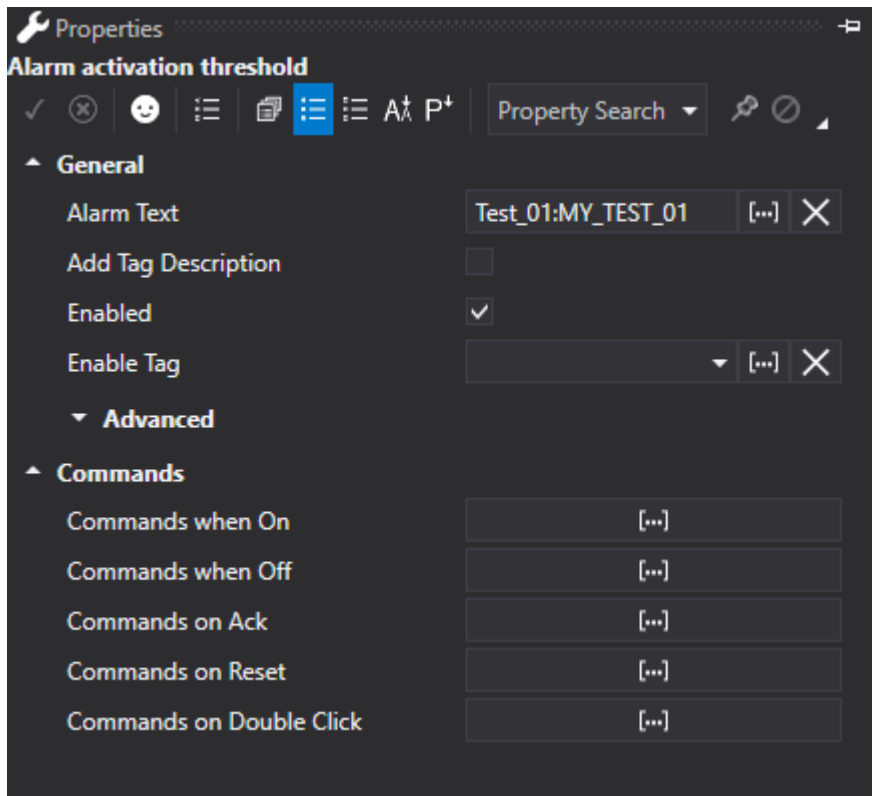


The Movicon.NExT Alarm Dispatcher is optional and therefore needs to be enabled on the runtime license.

In order to make sure that the project's Alarm Dispatcher functions correctly, you will need to configure it appropriately by defining the right server settings. Notification configuration means using (plug in) and defining messages to be sent as notifications.

# 12.2. Alarm Dispatcher Settings

To configure the Alarm Dispatcher, double click on the Alarm Dispatcher resource in the Project Window to open it in edit mode within the workspace. A window will display divided into three sections:



## General Settings

The following general notification parameters can be inserted in the Alarm Dispatcher's "General Settings" window. For further information on the Alarm Dispatcher properties, please refer to the chapter on "Advanced Alarm Dispatcher Settings":

### Application Name

The name of the Server application. This name will be used on Client browser lists to represent the OPC UA Alarm Dispatcher Server.

### Use a Custom Message

When this item is enabled, another 5 check boxes will open and which can be used to customize the text of the message to be sent by the Alarm Dispatcher. These check boxes also permit certain information to be removed from the standard message type that would otherwise be sent if leaving the "Use a Custom Message" disabled. This feature is very handy when using the Dispatcher's SMS plug-in in order to limit the characters used in sms to 160.

The message text can therefore be composed to include the following:

- Add Date/Time: adds the date and time of the notification (locale or UTC according to selection)
- Add Notification Name: adds the name of the dispatcher notification



- Add Alarm State: adds the alarm status (Active, Inactive, etc.)
- Add Notification Message: adds the 'message' associated to the notification
- Add Server Text (only for server notifications): adds the text of the alarm present in the ServerIONExT



Attention! In cases where all five check boxes to customize the message are left unchecked, the following warning message will appear::

*Beware! At least one option Of the Custom Message Definition should be checked, otherwise the standard message will be sent.*

Therefore the standard message format will be sent.

### Notify each state change

This property allows you to choose whether to send an alarm's notifications once only throughout the life span of its instance or send them at each transaction. Practically:

- When setting this property to True, each threshold transaction (ON, OFF, ACK, CONFIRMED) will generate the sending of a message.
- When setting this property to False, only one notification will be generated at the first occurrence of each transition of a threshold (ON, OFF, ACK, CONFIRMED). The cycle will only reset after the alarm has been Confirmed. For example, upon the activation of the alarm (ON), the message will be sent (Active Alarm). When the alarm passes from ON to OFF, another message will be sent (Alarm inactive). When the alarm is then acknowledged (ACK), the corresponding message will be sent (Alarm Acknowledged). At this point, however, if the alarm returns to the ON state and then OFF and ACK (even repeatedly), no further notifications will be sent. Only when the alarm is Confirmed, the message sending state will reset and the message will be sent again when the alarm is next activated.

### Use Local DateTime (Advanced)

When enabling this property the data and time in the event message sent by the dispatcher will be the Server's local time (therefore according to time zone). When left unmarked, in other words disabled, the date and time in the event message sent by the dispatcher will be in UTC format as indicated in the same message (i.e: Local Notification Name Active - UTC: 27/04/2017 08:31:52 - Local Alarm Active 01!).

### Manufacturer Name (Advanced)

The name of the manufacturer is entered here and will be published by the Server.

### Product Name (Advanced)

The name of the product is entered here and will be published by the Server.

### Product Address (Advanced)

This is used to enter a URL address referring to the product or the manufacturer, and this information will be published by the Server.

### Software Version (Advanced)

The product's software version is entered here and will be published by the Server.

### Build Number (Advanced)

The project's Build version is entered here and will be published by the Server.

## Transports

This window is used to define the protocols that are to be used for Server and Client communication. Once the Server has been started up, it will provide end-points according to the transports enabled in the project.



Attention! The "net.pipe" transport is a local transport which required both Client and Server to be started up in the same work session. This transport cannot be used if the Client project and Server project run on different machines.

## Status

This is a log window which activates when the Server is started up manually using the "Start Server" command and reports Server status or error messages.

## 12.3. Advanced Alarm Dispatcher Settings

A set of Advanced Settings for the Alarm Dispatcher is available and accessible through the Properties Window. These settings allow expert users to modify some of the detail settings that determine how the Dispatcher functions.

### Execution

#### Enable Log

This is used to enable/disable the creation of the LOG file.

#### Diagnostic Enabled

This is used to enable various diagnostic functions concerning the server and/or the communication drivers.

For further information regarding the Execution settings, please refer to the Data Server chapter on "Execution".

## Database Settings

#### Default Event Log Connection

This is the connection string that will be used by the Alarm Dispatcher to record the log's data. If no connection string has been set, the editor will open with the one for default. The default connection will be created in the SQL Server database using the local sql server instance and the Application Name as the name of the database.

If the database connection should disconnect for some reason while in runtime, the log's data will be flushed to the the xml file in the project's "..\AlarmDispatcher\Historian" folder.

#### Max. Event Age

This is used to set the maximum age for data in the DB.

#### Max. Concurrent Historian Accesses (Advanced)

This is used to set the maximum number of concurrent accesses to the server's historian side. Once this limit has been exceeded, subsequent access requests will be suspended until one of the accesses in progress has concluded.

#### **Historian Retry Waiting Time (Advanced)**

This is used to set the time in seconds to wait between one DB connection attempt and the next.

#### **Max. Historian Retries (Advanced)**

This is used to set the maximum number of historian access retries before flushing data in the cache to a local text file.

#### **Max. Records in Error (Advanced)**

This is used to set the max. number of records in error in the cache. Once this limit is exceeded, all records pending a new database entry attempt will be flushed to a text file in XML format in the local disk where the server is running.

#### **Min. Records in Cache (Advanced)**

This is used to set the minimum number of records in the cache pending database entry, below which will be flushed to a local text file in XML format.

#### **Max. Records in Cache (Advanced)**

This is used to set the maximum number of records in the cache pending database entry. Once this limit has been reached, the new records will be entered in the cache predisposed to flush data to local text files in XML format. In the event that this limit is reached in the second cache, the records will be irretrievably lost.

#### **Max. Record Deletions (Advanced )**

This is used to set the maximum number of records to be deleted from the database upon every maximum database age control check.

#### **Max. History Deletion Processes (Advanced)**

This is used to set the max. number of historical data deletion processes in the Database to be run simultaneously.

#### **Max. History Restoration Processes (Advanced)**

This is used to set the max. number of processes to be run simultaneously to restore historical data..

#### **Max. History Flush File Size (Advanced)**

Maximum size in MBytes of all Flush files. Once this limit has been reached, the oldest files exceeding this value will be permanently removed. This will be traced in the log with this message: "The maximum size of safe files allowed has been exceeded causing the permanent removal of some files from path '{0}' !" "

#### **Max. Alarm Events (Advanced)**

Sets the maximum number of alarm events to be loaded in the buffer.

#### **Max. History Continuation Points (Advanced)**

Maximum number of continuation points (as above) used for HistoryRead operations.

#### **Max. Request Age (Advanced)**

Maximum duration of incoming requests. Take note that the oldest requests are deleted.

## OPC UA SERVER Options

### Min. Publishing Interval

The minimum time in minutes to publish data supported by the server.

The server resets subscriptions that have a lower interval than this value. The value returned by the server in a subscription indicates when the server will update the value with the field value.

### Max. Publishing Interval

Maximum time in minutes to publish data supported by the server.

The server resets subscriptions that have a higher interval than this value. The value returned by the server in a subscription indicates when the server will update the value with the field value.

### Publishing Resolution

This field indicates the minimum difference in ms between the supported publishing intervals. This is always used to calculate the subscription interval which is returned by the server in respect to the one requested by the client.

### Max. Session Timeout (Advanced)

This is used to set the maximum time in ms with which a Client session will remain open in the absence of Server communication. It is advised to leave this set with the default value.

### Min. Session Timeout (Advanced)

This is used to set the minimum time in ms with which a Client session will remain open in the absence of Server communication. It is advised to leave this set with the default value.

### Max. Session Count (Advanced)

This is used to set the maximum number of client session that can be opened on the server.

### Max. Browse Continuation Points (Advanced)

This used to set the maximum number of continuation points to be used for browsing operations by clients towards the Server.

The "continuation point" is needed to perform a subsequent request to the server to retrieve remaining information. In fact, the amount of information received by the client might be incomplete due to a limited maximum amount of information that the client can receive for each individual interrogation.

### Max. Subscription Lifetime (Advanced)

Defines how long subscriptions can remain open without being published by client.

### Max. Message Queue Size (Advanced)

Defines the maximum number of messages saved in the queue for each subscription.

### Max. Notification Queue Size (Advanced)

Defines the maximum number of notifications saved in the queue for each monitored item.

#### Max. Notification Per Publish (Advanced)

Defines the maximum number of notifications saved in the queue for each publication.

#### Min. Metadata Sampling Interval (Advanced)

Defines the minimum metadata sampling interval.

## 12.4. Alarm Dispatcher Plugin List

The Alarm Dispatcher notification messages are sent using a specific communication tool called a Plugin. Each Plugin that has been made available for use will send messages according to the transport means provided.



The list of Plugins can be expanded by adding new ones that correspond to new and different means of communication to be used for sending notifications in the near future.

### Plug-In List

In the "Plugin List" window of the "Settings" section, the list of Plug-Ins already defined is shown. Through the menu Project > Alarm Dispatcher > Plugins it is then possible to add other Plugins among those available.

Double-clicking on an already existing plugin will open a pop-up window to define its characteristics. Each Plugin will have their own setting which will differ from those of others. The plugins currently available are:

- **Smtp Mail Sender**
- **GsmSMS Sender**
- **VOIP Sender**
- **Telegram Sender**

For further information on Plugin configurations please refer to the dedicated paragraphs: "Smtp Mail Sender", "GsmSMS Sender" and VOIP.

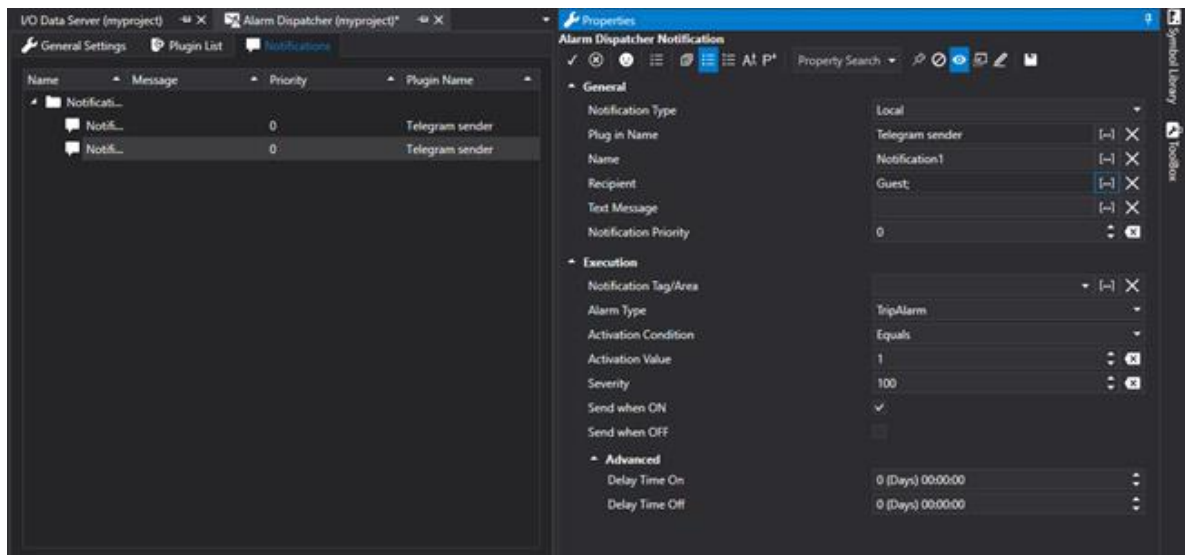
## 12.5. Alarm Dispatcher Notifications

The Notification properties in the Alarm Dispatcher's configuration correspond to the settings used for sending emails to predefined recipients:

### Notification Properties

The list of Notifications that the Alarm Dispatcher has to manage, by sending them to recipients according to the Plugin type used, is shown in the "Notification" window of the "Settings" section. New notifications can be added by using the "Add Notification" command. The Notifications can be organized in folders which are created using the "Add New Folder" folder.

When a new Notification is added, a popup window will open through which to define its characteristics. The same window also opens when double clicking on an already existing Notification.



*Window used for setting the Dispatcher Notification properties*

The Dispatcher Notification properties are as follows:

### Notification Type

- **Local:** A notification will be sent to the user when the alarm activates.
- **Server:** This allows you to select an ServerIO's Alarm Area or Alarm Source to receive notifications of the alarms in that Area or Source from the ServerIO. As a difference to the previous method, the specifics of each Alarm will not need to be redefined each time. In this case, the alarm's text will be passed to the Dispatcher directly from the ServerIO.



The properties displayed afterwards will vary according to the option chosen in this field.

### Plug-In Name

This is used to set the desired Plug-In type.

### Name

Represents the name associated to the Dispatcher Notification. The name must be unique within each single folder. When using the SMTP plugin for example, the name of the notification will also appear as the email's "subject". When using the SMS plugin, the name will be shown at the beginning of the message.

### Recipient

This field is used to select the user or user group name to send notifications to. When selecting a user group, the notification will be sent to all users of that group.

A message will be sent to the Chat ID of each user of User Group recipients of Telegram notifications. If several users have the same Chat ID (e.g. a number of project users belonging to the same Telegram chat group) only one notification will be sent to the whole group with the same ChatID.

If the recipient is one individual user, a message will be sent to the ChatID of that individual user selected.

### Text Message

The notification text to be sent is entered in this field.

For example, If you are using the SMTP plug-in, this text will be the body of the mail to be sent. If you are using the SMS plug-in the will part of the message text.



In cases of multi-language projects, the text can be a String ID to the sent message changes according to the set Default Language for recipient User/User Groups. In this case, you will need to make sure that each user or user group is set with a default language.

### Attachments

This field is only available when the SMTP plugin has been selected and is used for selecting one or more files to attach to the email.

### Notification Priority

This option is used for setting the notification with a priority. A delay time can be established by using this priority for sending the notification according to the Dispatcher's general "Priority - Delay Table" settings.

### Deviation Type

This selection field is available only when the ExclusiveDeviation, NonExclusiveDeviation, ExclusiveRateOfChange or NonExclusiveRateOfChange alarm type has been selected. For further information please see section: "Alarm Definition Settings".

### Enable High High Limit

This selection is used for enabling the "High High Level" alarm's intervention threshold. Once the threshold has been enabled, it will then be possible to insert the activation value using the "High High Level (Value)" field.

### High High Limit (value)

This field is only available when the ExclusiveDeviation, NonExclusiveDeviation, ExclusiveRateOfChange or NonExclusiveRateOfChange alarm type has been selected and when the threshold has been enabled using the "High High Level (Enable)" property. For further information please refer to the section on: "Alarm Definition Settings".

### Enable High Limit

This selection is used for enabling the "High Level" alarm's intervention threshold. Once the threshold has been enabled, it will then be possible to insert the activation value using the "High Level (Value)" field.

### High Limit (value)

This field is only available when the ExclusiveDeviation, NonExclusiveDeviation, ExclusiveRateOfChange or NonExclusiveRateOfChange alarm type has been selected and when the threshold has been enabled using the "High Level (Enable)" property. For further information please refer to the section on: "Alarm Definition Settings".

### Enable Low Limit

This selection is used for enabling the "Low Level" alarm's intervention threshold. Once the threshold has been enabled, it will then be possible to insert the activation value using the "Low Level (Value)" field.

### Low Limit (value)

This field is only available when the ExclusiveDeviation, NonExclusiveDeviation, ExclusiveRateOfChange or NonExclusiveRateOfChange alarm type has been selected and if the threshold has been enabled using the

"Low Level (Enable)" property. For further information please refer to the section on: "Alarm Definition Settings".

#### **Enable Low Low Limit**

This selection is used for enabling the "Low Low Level" alarm's intervention threshold. Once the threshold has been enabled, it will then be possible to insert the activation value using the "Low Low Level (Value)" field.

#### **Low Low Limit (value)**

This field is only available when the ExclusiveDeviation, NonExclusiveDeviation, ExclusiveRateOfChange or NonExclusiveRateOfChange alarm type has been selected and when the threshold has been enabled using the "Low Low Level (Enable)" property. For further information please refer to the section on: "Alarm Definition Settings".

#### **Time Unit**

This selection field is only available when a ExclusiveRateOfChange or NonExclusiveRateOfChange has been selected. For further information please refer to "Alarm Definition Settings".

## **Execution**

#### **Notification Tag/Area**

When the "Local" notification type has been selected, you will be able to select an I/O Data Server Tag to associate to the local alarm event to which the notification is associated.

When the "Server" notification type has been selected, you will be able to select an Area or an I/O Data Server Alarm source to which to associate the notification.

#### **Item**

This option is used for selecting the Server Tag based on which value the notification will be sent.

#### **Alarm Type**

This option is used for defining the alarm type to be used for sending the notification. The options are:

- **ExclusiveLevel**
- **NonExclusiveLevel**
- **ExclusiveDeviation**
- **NonExclusiveDeviation**
- **ExclusiveRateOfChange**
- **NonExclusiveRateOfChange**
- **TripAlarm**

#### **Activation Condition**

This property is only available when "Local" has been selected as notification type and a TripAlarm type alarm has been selected. For further information, please refer to the section on "Alarm Definition Settings".

#### **Activation Value**

This property is only available when "Local" has been selected as notification type and a TripAlarm type alarm has been selected. For further information, please refer to the section on "Alarm Definition Settings".



### Severity

This property allows you to set the notification with a severity level. The severity is a field that is displayed in the Historical Log in the recording of the event.

### Send when ON

When enabled, the alarm message is sent when the event's notification is ON.

### Send When OFF

When enabled, the alarm message is sent when the event's notification is OFF.

### Delay Time On (Advanced)

This parameter determines the notification's activation delay time. For further information please refer to "Alarm Definition Settings".

### Delay Time Off (Advanced)

This parameter determines the notification's deactivation delay time. For further information please refer to "Alarm Definition Settings".

### Notify Item

The selection made in this field will change according what "Notification Type" has been set as follow:

Local Notification Type: A ServerIO tag can be selected in this field and based on its value an alarm notification will be sent according to the predefined settings.

Server Notification Type: An ServerIO Area or Source from which to receive Alarms can be selected in this field. In this case notifications of Alarms triggered by the ServerIO will be sent directly.

## Commands for inserting a new Dispatcher Notification

The commands for editing Notifications can be found in the Alarm Dispatcher's Toolbar and are described below.



### Add New PlugIn

This command inserts a new PlugIn for sending Alarm Dispatcher Notifications.

### Add New Folder

This command creates a new Folder within which the Alarm Dispatcher's Notification can be inserted. These folders are only used for project organization purposes and can be nested at different levels.

### Add New Notification

this command is used for inserting a new Alarm Dispatcher Notification that will then need configuring as required.

### Start Server

The Alarm Dispatcher is in fact a OPC UA Server, that connects as Client to the Project's Server. The Start command is used for starting up the Server manually.

## Stop Server

This command Stops the Alarm Dispatcher Server manually after the Start command has been used for starting it up.

## 12.6. Plugin

### 12.6.1. Telegram Sender

The "Telegram sender" plugin allows you to send instant telegram messages similar to Whatsapp. Unlike WhatsApp, Telegram (<https://telegram.org>) is an open source messenger service based on cloud with instantaneous synchronization that allows you to access messages from different devices simultaneously, such as smartphone, tablet and computer. The Telegram client is available as:

- app for Android, iOS and Microsoft mobile devices,
- desktop client for Windows, Linux and MacOS
- a web app



il plug-in Telegram per Alarm Dispatcher non è compatibile con il sistema operativo Windows 7

In order to send messages you will need to create a Bot in Telegram which will be the recipient of messages sent by the Alarm Dispatcher.

The Bot will send on the messages it receives from the Alarm Dispatcher to the chat or group specified in the project Users' Telegram ChatID properties in the project.

The developer is responsible for creating a group chat in Telegram by adding all the notification message recipients, or for creating private chats between the Bot and each individual user.

Communication is one-way, therefore the Alarm Dispatcher communicates the activation of an alarm to the Bot which then sends the alarm activation notification to the various chats set in the project. As this type of communication is unidirectional, users will not be able to use the Bot to talk with each other but use private chats to contact each other outside the Telegram Group.

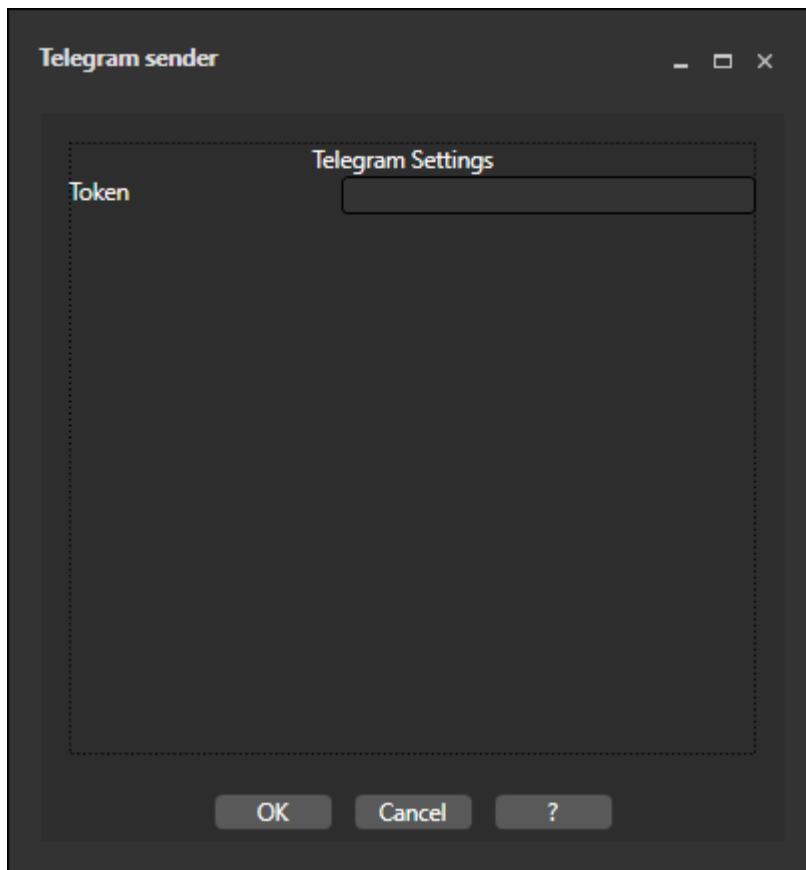


In order to use a Telegram Group to receive notifications from the Alarm Dispatcher, the Group will need to include a Bot.

Bots use Tokens in order to be identified.

In order to create a chat with a certain Bot, simply create a new chat and search for the Bot by its username.

The Plug-in configuration parameters are:



### Token

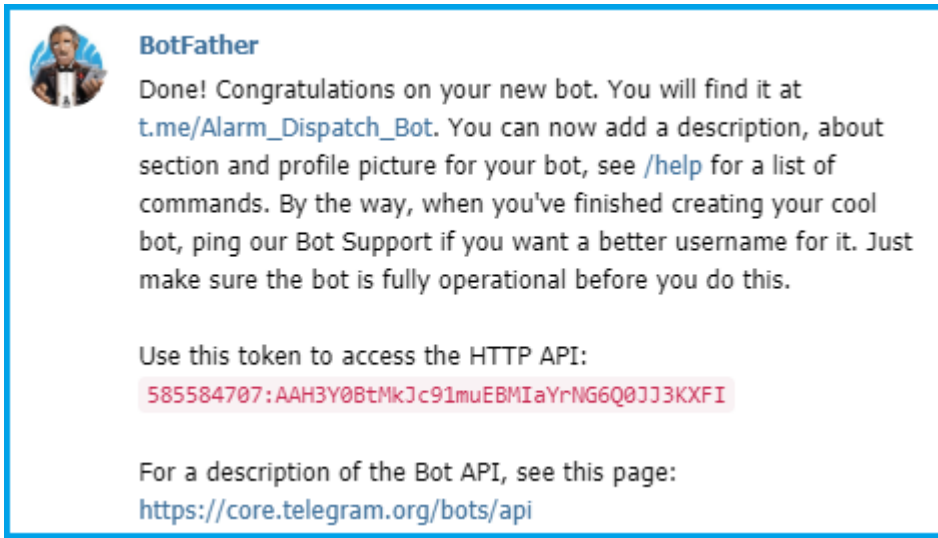
This parameter is used to identify the Bot to which messages will be sent.

In order to obtain this identification you will need to create a Bot using the Telegram app as follows:

- start up the Telegram app
- create a new message and use the magnifying glass icon to search for the 'BotFather' user among the contacts



- select Start to start the chat
- send a message with the "/newbot" text to command the creation of a new Bot
- define the Bot's name
- define the Bot's user name
- upon terminating the procedure, the token string will show to allow you to insert in the plug-in's property



After configuring the Token parameter, you will need to retrieve the ChatID of the Chat which the Bot will send notifications to.

The Chat ID should be indicated in the User's properties defined in the project, and the Recipient parameter should also be set correctly in the "Alarm Dispatcher Notifications" property group.

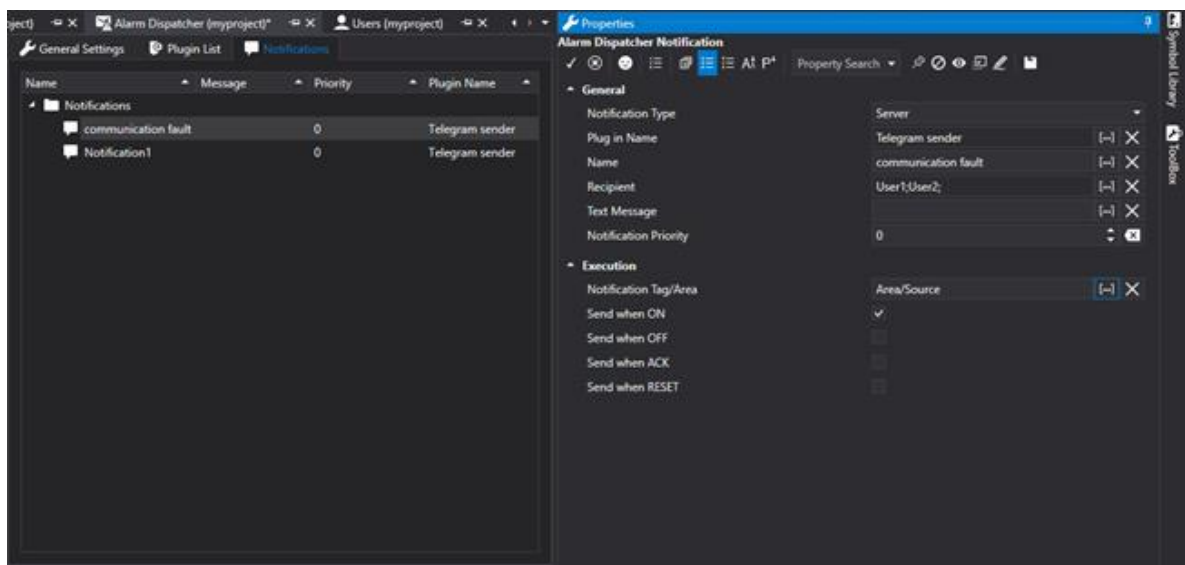
## Configuration for sending notifications to individual users:

After having configured the Alarm Dispatcher' Bot in Telegram and its Token in Movicon.NExT, you will need to define the notification recipients.

When needing to notify project users individually, because they belong to different Groups or because you wish to notify each user independently for instance, you will need to follow the steps described below.

### Configure the Alarm Dispatcher's notification

The first thing to do is configure the Alarm Dispatcher's notification so that the recipients are individual project users (e.g. "User1" and "User2") as shown in the screenshot below:

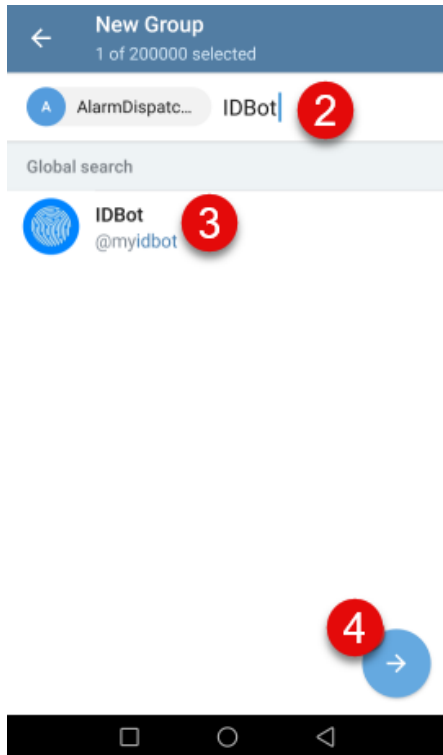


### Recovering User ChatID in Telegram

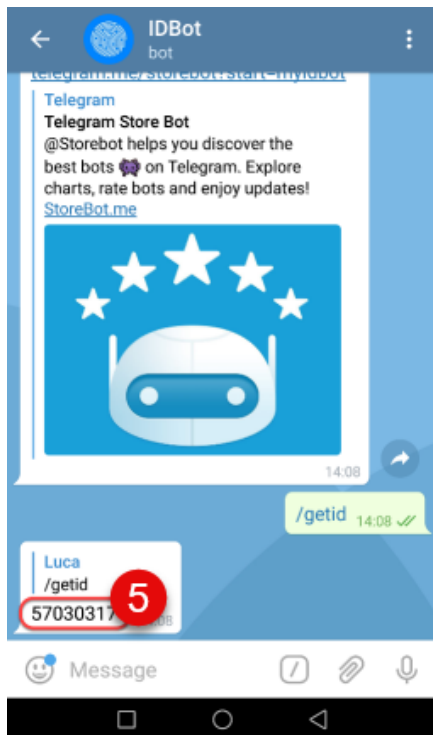
A user's ChatID in Telegram can be recovered using a Bot called IDBot as described below:

1. Access Telegram.

2. Use the search box (magnifying glass icon) to search for "IDBot"

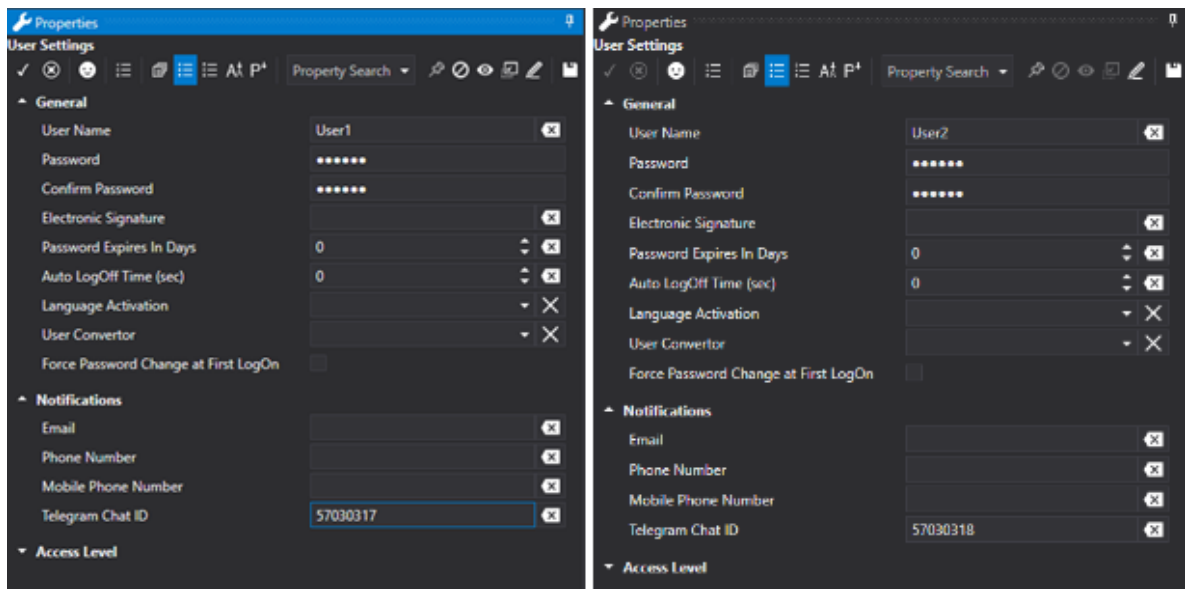


3. Select the "IDBot" Bot.
4. Start chat.
5. Type in the "/getid" command to obtain your assigned ID.



### Movicon.NExT project user ChatID configuration

The string received from the IDBot for each user should then be copied in the Movicon.NExT project user's "Telegram ChatID" property as shown below:



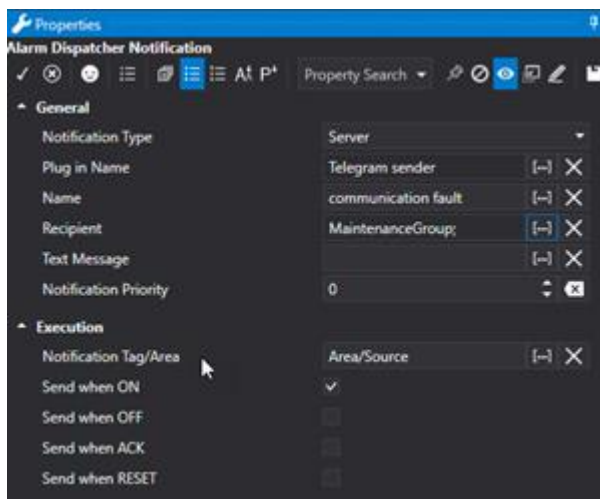
## Configuration for sending notifications to User Groups:

There are two different methods to use for sending the one only notification to a User Group:

- Configure the Alarm Dispatcher's notification so that the recipient is an individual "virtue" project user to which the ChatID of a Telegram Group will be associated.

**i** In this way, those interested in receiving notifications can make a request to belong to the Telegram Group without needing to edit the Movicon.Next project.

- Configure the Alarm Dispatcher's notification so that the recipient is a User Group of the project (e.g. "MaintenanceGroup"):



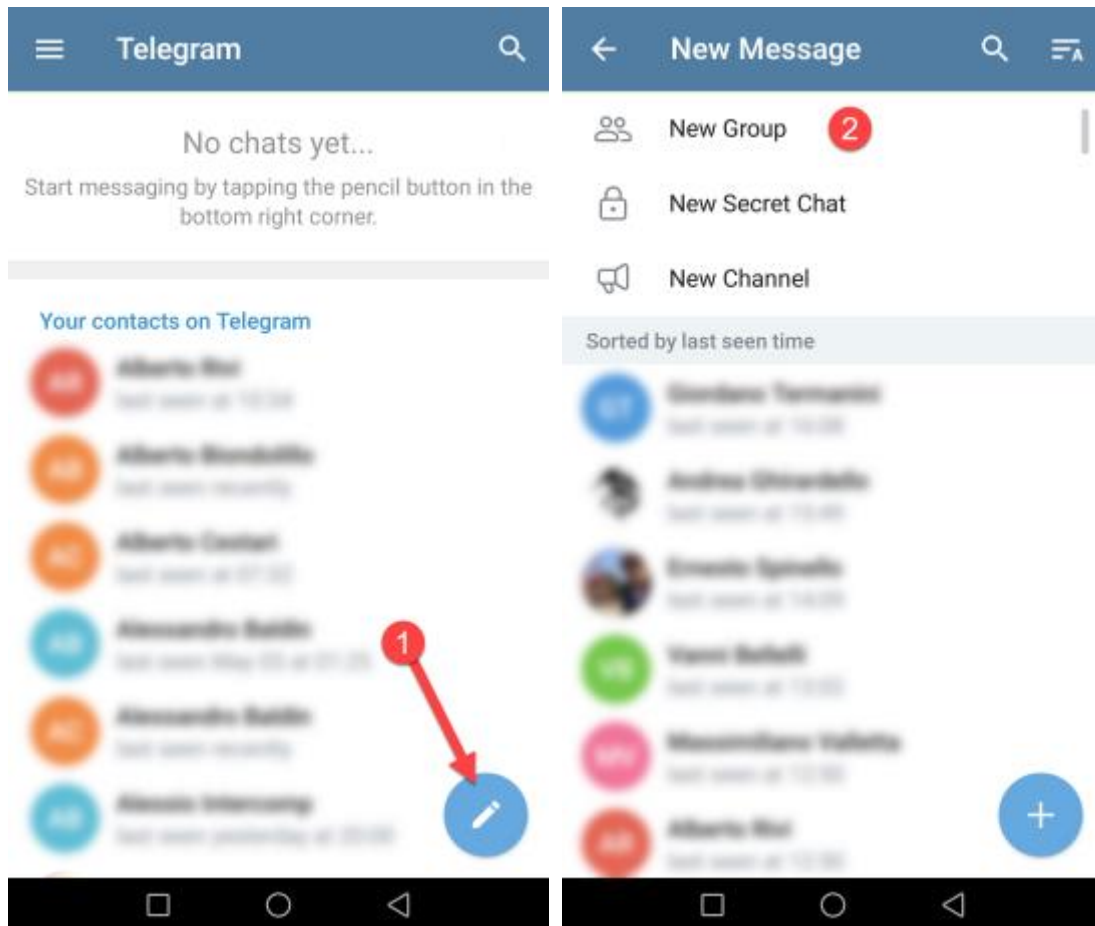
then associate the same Telegram Group's ChatID to all the users of the Movicon.NEXT User Group.

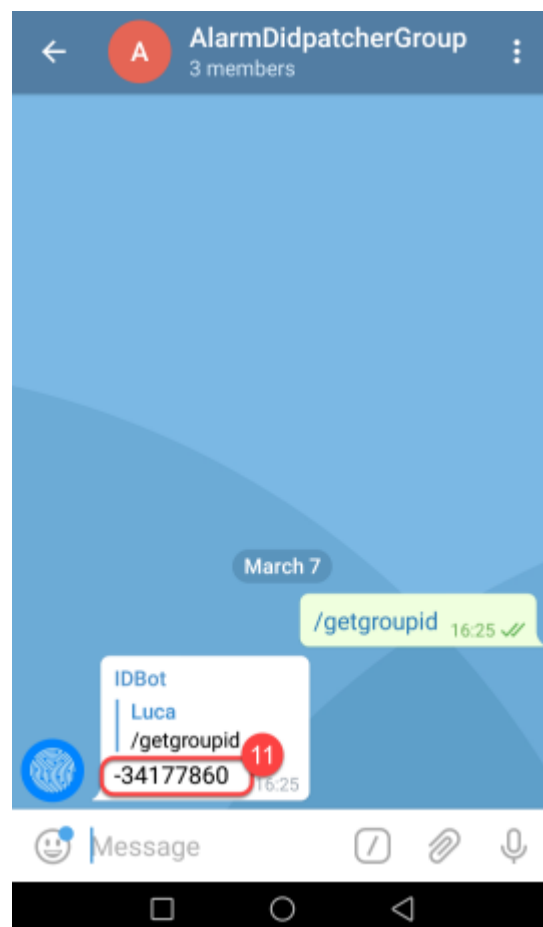
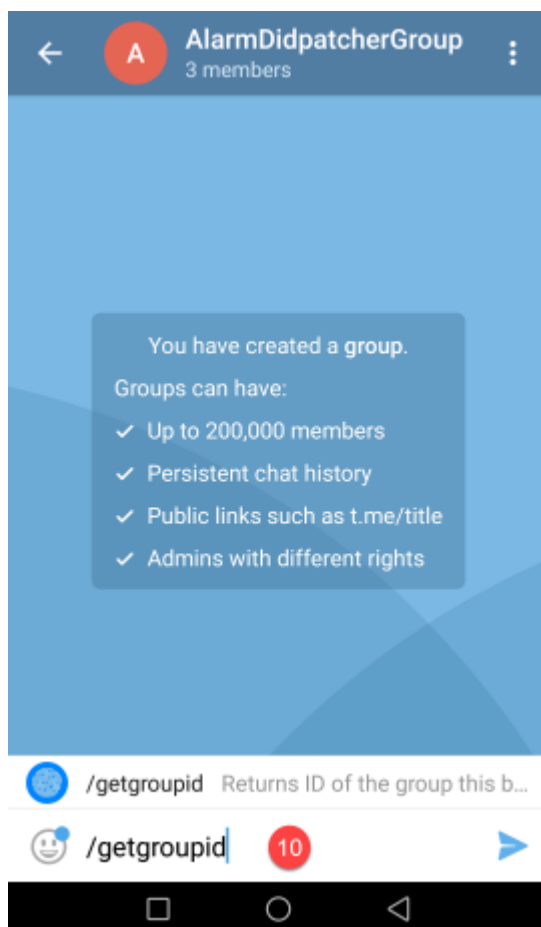
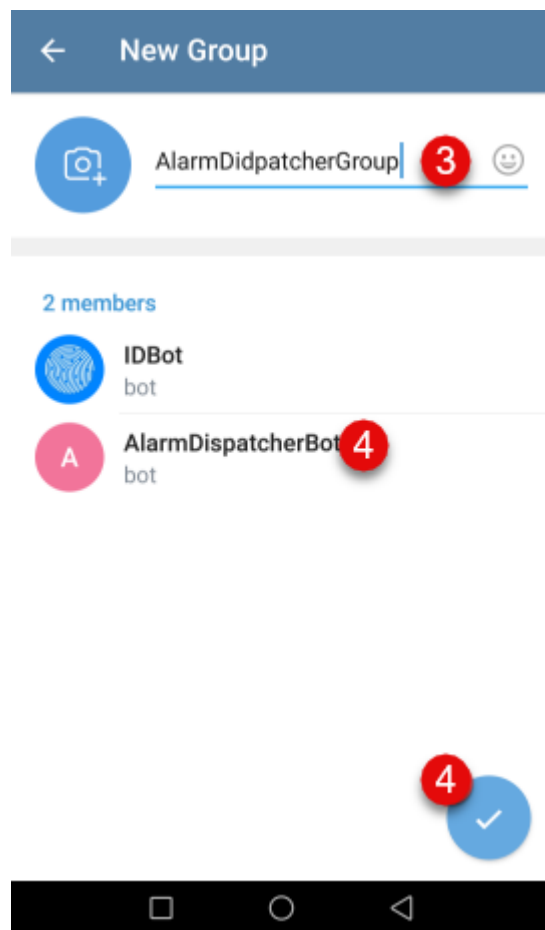
### Configure a Telegram Group and recover the ChatID

The ChatID of a Telegram Group can be recovered using a Bot called IDBot Bot by using this procedure:

1. Access Telegram to start a new conversation.
2. Select "New Group".

3. Type in the name of the previously defined Alarm Dispatcher's Bot, "AlarmDispatcherBot" in this example, in the search box.
4. Select it to added it to the group.
5. Type in the "IDBot" name in the search box.
6. Select to add it to the group.
7. Confirm the creation of the group.
8. Type in the name of the Group.
9. Confirm.
10. Once the group has been created, type in the `/getgroupid` command.
11. The "IDBot" Bot will respond with the group's ChatID.





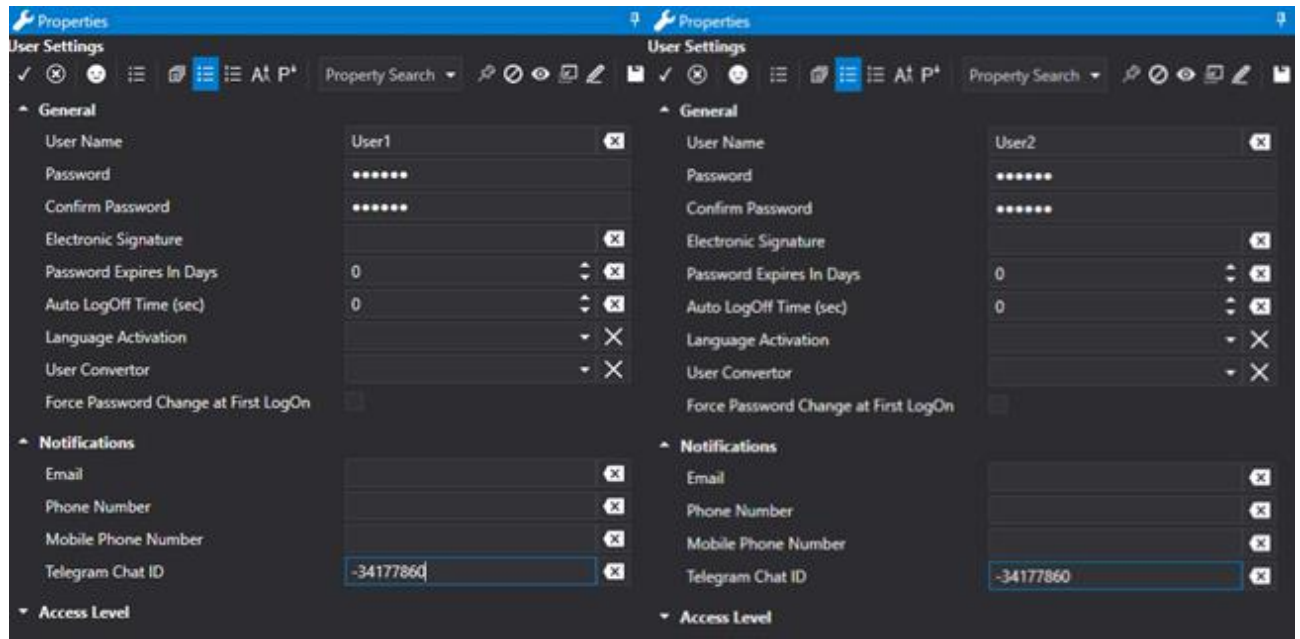




Once the ChatID has been obtained, the "IDBot" Bot can be removed from the group. The "AlarmDispatcherBot" Bot, however, is always kept in the group.

### Configuring ChatID for Movicon.NExT project users

The ChatID string of the Telegram Group received from the IDBot should be entered for each user of a Movicon.NExT User Group:



### 12.6.2. Smtip Mail Sender

The "Smtip Mail Sender" plugin is used for sending messages with email (electronic post) using direct access to a Server with SMTP protocol. In this case the sending of emails will not require an electronic post program (i.e Ms Outlook) to be installed in the system PC. However the notification system does require internet access.

**Smtplib mail sender** [X]

SMTP Settings

Server Address [ ] ?

Server Port 25

Username [ ]

Password [ ]

From [ ] ?

Enable RAS ☐

Dialup Entry name [ ]

Phone number [ ]

RAS User [ ]

RAS Password [ ]

Retry timeout (sec.) 20

Disconnect after (sec.) 20

Retries number 3

Custom Phonebook Path [ ]

[OK] [Cancel] [?]

*The "Smtplib Mail Sender" plugin settings window.*

The "Smtplib Mail Sender" plugin's configuration parameters are:

### Server Address

This setting is used for defining the post Server name (post service provider) with whom an account has been established.

### Server Port

This parameter allows you to specify the port to use (default value 25).

Depending on the port number used, the SMTP client will use the appropriate security type. The supported protocols are: SSL, TLS and STARTTLS.

### Username

This edit box is used for specifying the user name for accessing the post Server. This user name must correspond to an account that the Server can recognize.

**Password**

This edit box is used for specifying the password of the user who accessed the post Server.

**From**

This setting is used for defining an address to be specified in the message as the sender's address. This setting is useful when using Servers that only allow mail to be sent if the sender belongs to its dominion.

**Enable RAS**

This option enables the use of a RAS connection for accessing the Server via modem. Contrary to this, Server access (internet) will use a permanent connection (e.g. LAN).

**Dialup Entry Name**

This parameter is used for specifying the name of an already available connection in the operation system (previously inserted as a network connection) that can be used.

**Phone Number**

This parameter is used for defining the telephone number for accessing service in cases where the Dialup Entry Name is not used.

**RAS User**

This parameter is used for defining the name of the user for accessing the service in cases where the Dialup Entry Name is not used.

**RAS Password**

This parameter is used for defining the password of the user accessing the service in cases where the Dialup Entry Name is not used.

**Retry Timeout (sec)**

Time to wait before retrying to connect when in error.

**Disconnect After (sec)**

Time after which disconnection will take place.

**Retries Number**

Number of connection retries when in error.

**Custom Phonebook Path**

This parameter is used for specifying the Windows "Phonebook" file path from which to retrieve connection information. In this case the name of the connection to be used must be specified in the "Dialup Entry Name" parameter.

**12.6.3. Sending Vocal Messages**

Vocal messages can be sent to an individual recipients or a group of recipients by using the VOIP system. The message is considered listened to and can no longer be resent when the recipient answers the call and confirms to have listened to the message or messages when pressing the phone buttons (see "Imposta forzatura ACK" property). Several messages may be sent in addition to the original one while answering the call. This is because when the first message is being listened to, the plugin searches for

other messages addressed to the recipient who is listening to the message. When other messages addressed to the recipient who is currently listening to the first message, they will be sent straight away to avoid calling same recipient again.

Voice over IP message sender

Title

Welcome Message

Farewell Message

Next Message

TimeOut 60

Max Retry 3

Notify All ☐

Force Sharp ACK ☐

Enable SIP Server ☐

Registration Required ☒

Display Name

User Name (SIP)

Register Name (SIP)

Register Password (SIP)

Registration TimeOut 120

Host

Port 5060

Selected Voice

Pronunciation Rate 0

ACK Server Alarm ☐

OK Cancel ?

Calls can be made in two different ways:

- **Direct IP Call:** Here the recipient is a VOIP client and can be reached by specifying their IP address. This does not require any account from a VOIP service provider.
- **Call with provider:** Here the services of a provider are used and communication is setup with the plugin using a pstn telephone and mobile phone network. In this case you will need to open a SIP account with the VOIP service provider and get access to the server. The server access settings are included in the plugin's configuration.



Skype was used as the VOIP service provider when testing Movicon.NExT. Please consult the relevant provider information on subscription fees and restrictions.



To install a different language for the Windows voice syntax please consult the relative pages in Help di Microsoft.

The "Voice over IP message sender" Plug-in configuration parameters are as follows:

#### **Welcome Message**

Synthesized welcome text when recipient answers call.

#### **Farewell Message**

Synthesized text used before terminating call.

#### **Next Message**

Synthesized text when additional alarm messages are sent in addition to the first one sent in the same call. This text is heard between one message and the next.

#### **TimeOut**

Timeout in seconds for operations during the call.

#### **Max. Retry**

Maximum number of call attempts towards recipient.

#### **Notify All**

This concerns calls to recipient groups. When set to true: if the message is listened to by one of the group members, it is only acknowledged after the message is sent to all the other group members. When set to false (default): if one group member listens to the message, it is acknowledged as being sent and therefore no other calls will be made to the other group members.

#### **Force Sharp ACK**

This concerns the mode used by the recipient to confirm to have received and listened to message. When set to false (default), the recipient need only press one of the phone keys (0-9, \*, #) to confirm after which the plugin starts synthesizing the message. When set to true, the recipient must press the # key to confirm that they have listened to message. If the recipient presses another key, the plugin will send the text of any other existing messages (addressed to the same recipient), otherwise it will repeat the same message again. When the recipient presses #, all existing messages addressed to that recipient will be acknowledged as listened to even though not yet sent by the plugin.

#### **Enable SIP Server**

When set to true, the plugin will try to connect to a VOIP provider using the settings detailed below. When set to false, the plugin will try to connect to the recipient using a direct IP call.

#### **Registration Required**

(SIP Server) When set to true, the telephone line is registered once connected to the VOIP server (SIP account may then be required).

#### **Display Name**

(SIP Server) SIP account settings.

#### **User Name**

(SIP Server) user name for VOIP server authentication.

**Register Name**

(SIP Server) Name used by server for registration. The account often uses the same string for DisplayName, UserName and RegisterName.

**Register Password**

(SIP Server) password for VOIP server authentication.

**Registration Timeout**

Timeout in seconds used for confirming registration to VOIP server.

**Host**

(SIP Server) Url (or IP address) of VoIP server.

**Port**

(SIP Server) TCP port to connect with the VoIP server (5060 is used for default for the SIP service).

**Selected Voice**

Name of the voice to use for the vocal syntax. The combo box shows all the voices that have been installed on the pc and which the TextToSpeech service can use.

**Rate**

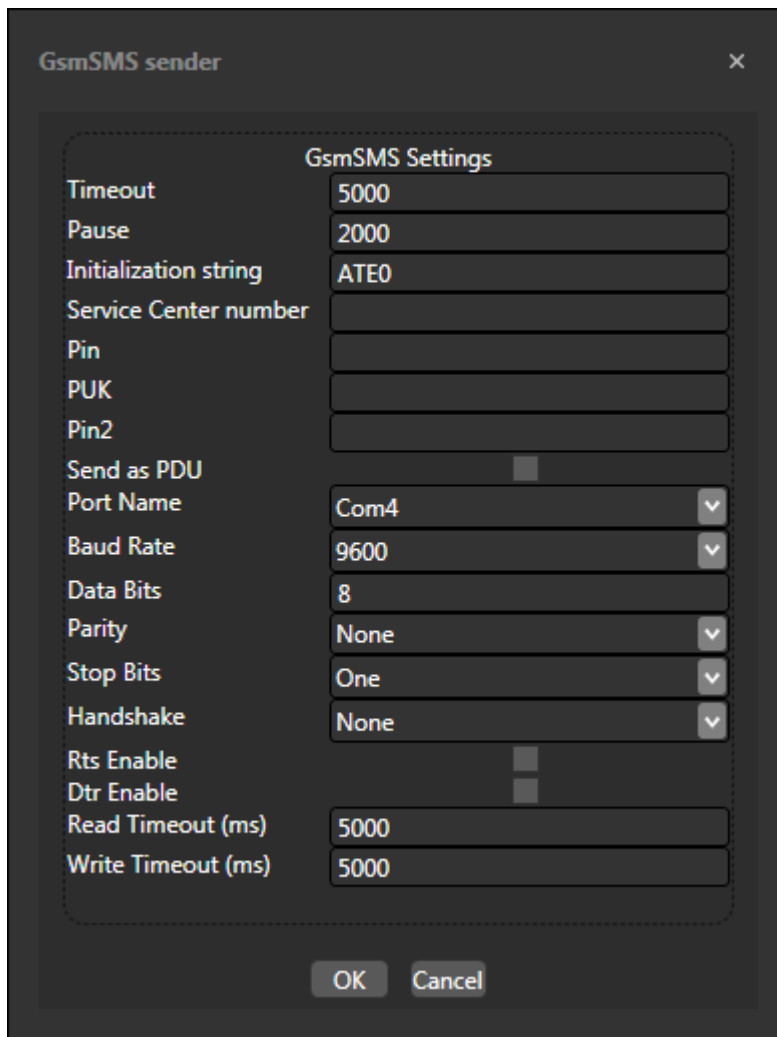
TextToSpeech setting. Range -10,+10. Influences the speed, fast or slow, with which the text is synthesized.

**ACK Server Alarm**

When this is enabled, the alarm will be silenced on the server side, passing from the unacknowledged state to the unconfirmed state, until the notified user has completely listened to the related messaged.

**12.6.4. GsmSMS Sender**

The "GsmSMS Sender" plugin enables messages to be sent with the Gsm-based SMS technology using any standard GSM modem.



*The "GsmSMS Sender" plugin settings window.*

The "GsmSMS Sender" plugin's configuration parameters are:

#### **Timeout**

This parameter is used for specifying the Timeout duration to be used by the serial port to which the Modem is connected.

#### **Pause**

This parameter is used for specifying the Pause time to be used by the serial port to which the Modem is connected.

#### **Initialization String**

ATE0 is normally used as the modem's initialization string but as a precaution, please refer to the documentation of the modem being used.

#### **Service Center Number**

This number may be required by the SIM Card used by the modem for accessing telephone operator service centers.

#### **Pin**

The Pin of the SIM card inserted in the Modem.

**PUK**

PUK of the SIM card inserted in the Modem.

**Pin2**

Pin2 of the SIM card inserted in the Modem.

**Send us PDU**

When this option is enabled the message will be sent in PDU format.

**Port Name**

Name of the serial port to which the Modem is connected.

**Baud Rate**

Baud Rate of the serial port to which the Modem is connected.

**Data Bits**

Data bit number of the serial port to which the Modem is connected.

**Parity**

Parity of the serial port to which the Modem is connected.

**Stop Bits**

Stop Bit

**Handshake**

Handshake type of the serial port to which the Modem is connected.

**Rts Enable**

Enables the Rts signal of the serial port to which the Modem is connected.

**Dtr Enable**

Enables the Dtr signal of the serial port to which the Modem is connected.

**Read Timeout (ms)**

Timeout for reading messages.

**Write Timeout (ms)**

Timeout for writing messages.



# 13. Users & Security

## 13.1. General Security Concepts

The Movicon.NExT project security management ensures both project developers and users the maximum protection with security system access management that complies to the strict regulations and standards.

The platform's security system has two precise data protection criteria:

1. Project Protection
2. User Authentication and access to controls and data in runtime

### Protecting your project

A Movicon.NExT project can be protected and encrypted to enable the developer to protect their know-how and prevent project access and modification by unauthorized persons. Project protection in edit mode is totally distinct from the user management to access controls in runtime. For example, a project can be protected and encrypted without activating the users management in runtime.

### User and Password Management for Operators

The User and Password management offers the option to manage access to project controls and functions during runtime according to the security standards. The developer can set for all the functions and controls, where necessary, with user authentication request by Logging in according to access level and/or area level.



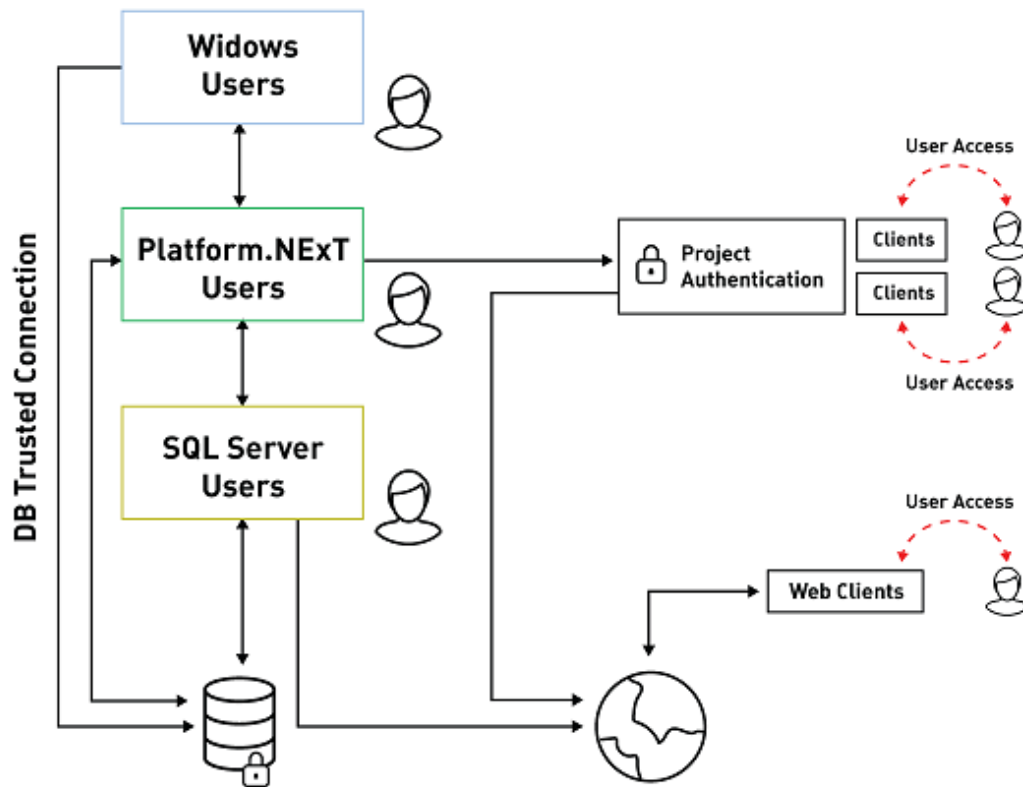
**ATTENTION!! It is important that the developer or administrator adopt ways to remember or retrieve a forgotten password. Once a password has been lost it will be impossible to gain access!**

**If this should ever happen please contact the Customer Support Services.**

## 13.2. Security Management

The project security management in Movicon.NExT is based on the ASP.NET. Membership Providers model. In addition to ensuring the maximum security policies for managing users, it also offers the use of "providers", so that the security model (data source) is independent from the authentication system.

The Membership is an API used to manage users within a NET. application to permit user authentication management. The Membership configuration within the Microsoft .NET ambient is made possible due to the Providers which connect applications to authentication system. By using these Providers, a Movicon.NExT project can guarantee the same level of security despite any authentication system customization that might be based on different databases or on authentication systems based on biometrics readers or scanners, RFID (Radio Frequency Identification), Badge or others for instance.



*This diagram shows an example of the Movicon.NExT default user security and authentication concept*

## Authentication System

According to the Movicon.NExT security model **when the User and Password Management is enabled** each sensitive project control or function might require user authentication and access level verification in order to grant users permission to use them. This usually happens the Client side where command executions, data modifications such as set-points, entry-points, recipes or other require interaction with the user interface. However the security management may also be provided on the Server Side to accept or deny data modifications or variations.



*This is an example of command access protection and the authentication and permission process*

In whatever case, the project commands and functions may nevertheless be subject to user access level and privilege verification after the user has been authenticated.

In addition to enabling the Users Management, this also requires that users be entered either in edit or runtime mode to enable them to activate commands according to their hierarchical level.

Therefore there is a project user profile/credentials that is/are automatically mapped to the SQL Server database which used the Membership Provider for user authentication. when enable the platform's authentication system requires that:

1. the "Users Manager" properties be assigned to all graphical objects and functions to be protected by establishing User access level and area.
2. the User profiles have been entered with the relative User Level property (hierarchy level) and access area.
3. the Membership Provider has been configured correctly. The system is configured with the Microsoft SQL Server Provider for default.

## Assigning command access criteria

Assigning the access policies to the project's functions is generally done on the **Client** side in control objects or functions managed in screens. For example, a "Start" button may set with a certain user level in its properties. This can be done for any object, symbol or function that is used for command executions. However, security policies can also be defined on the **Server**, and inherited by each associated object hierarchically.

For example, a Tag in the Server can be set with an access level so that every time its value is changed by a user from elsewhere, the user will be asked for User authentication beforehand.



It is the job of the programmer to establish where the user authentication is to be associated. Normally such procedures are established on the Client side but it would be very convenient to use these procedures to protect Server data in distributed or more complex architectures.

Independently from where the security policy is used, it must be set in the interested object's properties which are those grouped under "**User Management**" if object is on the Client side or under "**Access Level**" if on Server side.

Therefore please refer to relevant topics describing the use of these properties.

## User Privileges

In addition to user authentication, managed with Memberships, each user can be set with specific privileges by means of using their properties.

The settings of the user privileges are bases on the following criteria:

### Level

This is a hierarchical level which is defined by using a value to establish the user's level of access. The lowest level value indicates that the user has the least privileges. Access to the control will then be accepted if the level requested is equal to or higher.

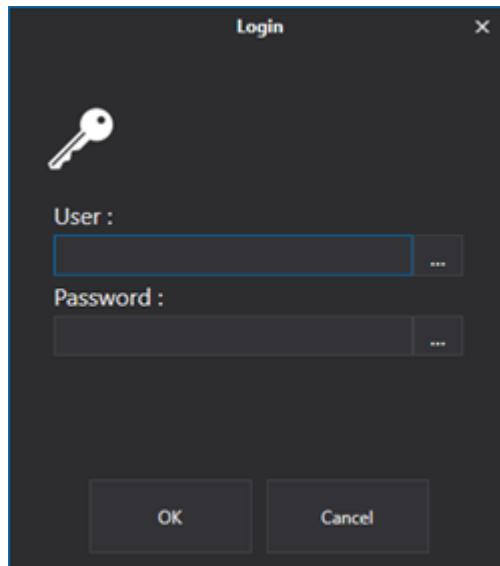
### Access Mask

In addition to the hierarchy level, users can be assigned one or more "Access Areas" by using the "Access Mask" which provides the possibility to select up to a max of 31 areas.

An object to which a specific Access Area has been assigned in Read will be only visible to those users who have been assigned with that specific Area (otherwise the object will not result visible). An object to which a specific Area has been assigned in Write will result enabled only to those users who have been assigned that specific Area (otherwise the object will result disabled showing semi-transparent). All areas are enabled for default for each user.

## User Log On

The system will ask users to Log On wherever data has been protected with user authentication. The window below shows when user log on is needed. All successful Log Ons are recorded.



This image above shows the user authentication window for logging on. It is displayed by the system when the user attempts an operation that requires a certain user level and therefore user authentication. When the user logs on with a user level equal to or higher than the one requested by the system, he or she will be granted access.



The system traces all user Log On/Off dates and times in the System Log.

The following topic describe all the User and User Group properties and functions.

The Movicon.NExT security model is based on Membership Providers to ensure maximum security and independency from the type of authentication desired. By using Membership Providers, you will be able to customize the type of authentication system management you prefer best. Movicon.NExT uses Microsoft Membership Provider based on the SQL Server for default in total automatic and safety. The user can use the security provider with complete transparency as the SQL Server Express 2012 instance is automatically installed while installing Movicon.NExT if the SQL Server to connect to is not already installed. The "Movicon.Membership" database is created in the SQL Server Express 2012 instance along with tables to manage project users.



One of the advantages of using Providers is the possibility to share with other Clients, whether local or remote (e.g. Movicon.NExT Clients or Web Clients). In addition, the Provider can be changed and configured in the application's settings to allow the use of authentication systems other than the one used for default based on SQL Server.

### Membership Provider Customization

Whenever needed, the expert design engineer can change the Membership Provider they are using by replacing the Microsoft SQL Server provider with the one desired.

To setup a new Provider, you will need to modify the application's XML configuration file with the .config extension (Movicon.NExT.exe as Client or PlatformNExTIOServer.exe as Server).

In this file you will need to set the name of the Provider desired and the connection string data and parameters

In order to configure the desired Provider correctly, please refer to its documentation or, if necessary, contact Technical Support for further information.

## 13.3. Project Protection and Encryption

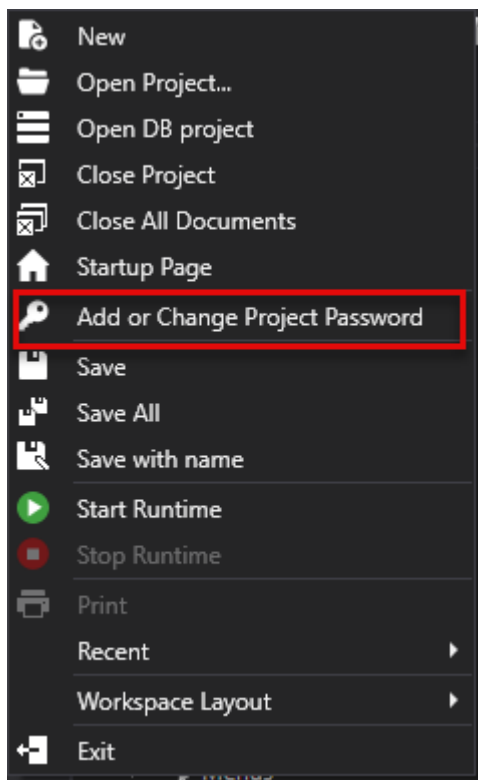
The Platform NExT development environment provides the developer a functionality that will enable them to protect their project against modifications by unauthorized users.

After having activated the protection, the project will be protected and encrypted to prevent any unauthorized access to the project in edit mode. The developer will then be the only one to allowed access to the project in edit mode in order to protect the project from being modified by third parties.



**ATTENTION!! It is important that the developer or administrator adopt ways to remember or retrieve a forgotten password. Once a password has been lost it will be impossible to gain access!**

**If this should ever happen please contact the Customer Support Services.**



To activate project protection simply access the File menu > Add/Change Project Password. This will display a window through which the desired password can be entered and confirmed. After the password has been confirmed, the project will be protected and encrypted. The password must be used when you next want to access the project in edit mode.



Project password and encryption is disabled when the project is saved to the database. Saving a protected project from a file to the Database requires that the protection be disabled before it can be saved.



When enabling the project security all the adhering files will be automatically encrypted and only accessible by using the password set in development mode.



Project security is not bound to user management activation and therefore the project can be protected without it having users.

## 13.4. Users and Passwords Management

### 13.4.1. General User Management Settings

The User Management has a series of General Settings that are displayed in the Properties Window after having select the general User and Group resource from the Users Editor.



The use of the Movicon.NExT User Manage implies that the data exchange between Client and Server takes place with the "Sign" security policy using Certificates.

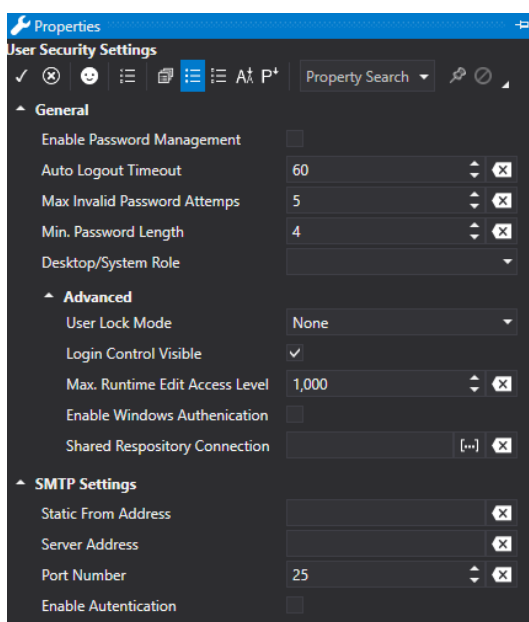
In this context, when the project includes distributed Client and Server architecture, the Server and Client will be required to exchange security certificated as indicated in "OPC UA Certificates "



In order to activate the User Management at Runtime, you will need to enable the "Enable Password Management" property.



The default user who is activated at Runtime startup is an Anonymous user that has access level 0 and no area access in read and write.



## General User Management Properties

Below are described the general properties of the project user management.

### Auto Logout Timeout

This field is used to enter a time in seconds which will be used for logging out the active user automatically. The time in seconds refers to the inactive time. Therefore a logged in user that is inactive for the time set here will be automatically Logged Out for security reasons. If the user reassumes activity they will need to log on again.

### Max. Invalid Password Attempts

This allows you to specify the maximum number of failed access attempts after which a user is blocked from using the system.



Setting the threshold criteria for user locks determines the number of failed login attempts that causes the system to block access of a specific user.

When a user is locked, they will no longer be able to access the access even when using the correct credentials. The locked user will be denied access until reinstated by using the appropriate commands provided by Movicon.NExT.

This functionality requires you to set a failed login attempt value and to specify which users are to be locked using the "**User Lock Mode**" property.

### User Lock Mode

This property gives you the possibility to specify which users are to be locked after attempting a certain number of failed logins.

This property come with the following options:

- None: the user lock mode is disabled (default value)
- OnlyEditableUser: Only those users with access levels lower or equal to the one set in the 'Max Runtime Edit Access Level' will be locked when exceeding the number of failed login attempts set in the 'Max. Invalid Password Attempts' property.
- All: In cases in which the number of failed login attempts is greater than the value set in the 'Max. Invalid Password Attempts' property, all users with different access levels will be locked out.



When needing to reinstate a locked user, use the commands contained in the 'Users' tab of any one of the Movicon.NExT objects.



Domain users are excluded from the lock users management as they are not managed directly by the Movicon.NExT membership provider.

### Enable Password Management

Enabling this property will activate the user manager during project Runtime. The programmer can then decide when to activate or deactivate the project's entire user management.

### Login Control Visible

When the user management is enabled in the project, this property allow you to enable/disable the displaying of the logo for the user Login/Logout at the top right of the startup page. This logo will only appear in the projects startup screen but not in all the other screens. Therefore, the logo will show in one of the following screens based on the project's startup mode:

- Tile Page

- Geo Page
- Startup Screen in "Main Screen" mode

### Max. Runtime Edit Access Level

This parameter is used to specify which maximum hierarchy level is permissible for a user who is inserted or changed at Runtime.

### Min. Password Length

The "MoviconNEXt.exe.config" system file can be modified to force the use of numbers and special characters in passwords by adding the "minRequiredNonalphanumericCharacters" variable. The inserted value, will be the minimum number of non-alphanumeric characters required in the password.

```
<membership defaultProvider="AspNetSqlMembershipProvider">
  <providers>
    <remove name="AspNetSqlMembershipProvider" />
    <add name="AspNetSqlMembershipProvider" type="System.Web.Security.SqlMembershipProvider, System.Web,
      connectionStringName="LocalSqlServer"
      enablePasswordRetrieval="false"
      enablePasswordReset="true"
      requiresQuestionAndAnswer="false"
      applicationName="/"
      requiresUniqueEmail="false"
      passwordFormat="Hashed"
      maxInvalidPasswordAttempts="5"
      minRequiredPasswordLength="1"
      minRequiredNonalphanumericCharacters="0"
      passwordAttemptWindow="10" passwordStrengthRegularExpression="" />
  </providers>
</membership>
```

### Desktop/System Role

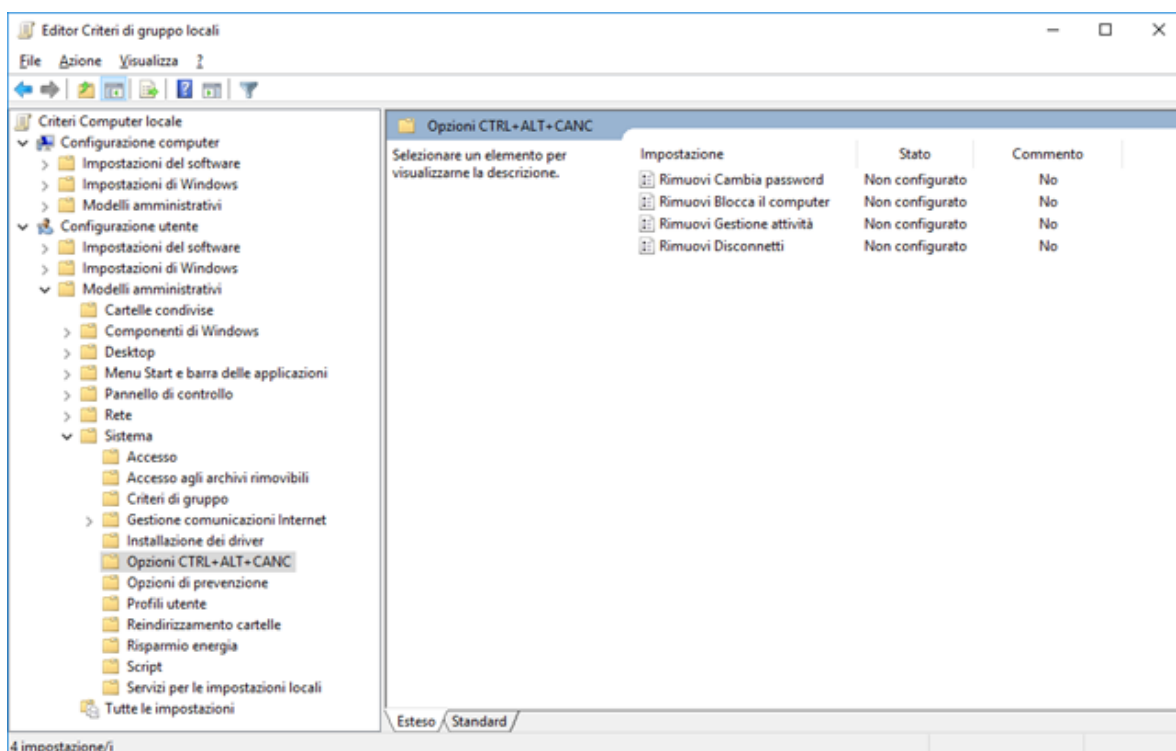
This property is used to associate a "Group" and then the relative Users with rights for executing system operation commands. The system commands are those commands that are not linked directly to the project but to the operating system's functions which perform those operations. For example, closing the application window, reducing to icons, desktop access and closing the application.



The system commands are not managed by the application's objects but by Windows OS. To enable correct user management, it is important that the programmer decides which user has rights to use those operating system commands which are accessible.

**Attention:** In addition, the "CTRL+ALT+CANC" combo keys cannot be blocked using this Movicon feature. In this case, the Windows' Local Group Policy Editor can be used to set these restrictions accordingly to prevent the anti-malware from intervening. To startup the editor, digit "gpedit.msc" in Cortana's search bar and press enter:





The safest configuration to use rests on the use of the Movicon Server started up as service in combination with the Windows settings indicated above.

## Enable Windows Authentication

When this functionality is enabled, the User Management system will allow authentication of those users registered in the Domain of the Windows OS in which the project is run. In this way, if the user management cannot find a user in the project user list, it will ask the OS for user authentication using the LDAP protocol with the 'domain\user' format.

If the user is recognized, they will be authenticated from the Domain, and then validated for using the project. In this case their access rights will be determined by their **group** membership.

When enabling this function, the User Management system allow the authentication of users registered in the local PC or the system domain in which the project is running in addition to the declared project users.

In this way if the user management cannot find a user in the project's user list, it will ask the operating system for authentication using the LDAP protocol.

To authenticate a local PC user simply insert the user's name and the PW (it must not be preceded by local\ or Hostname\).

To authenticcte a domain user, you will need to use the 'domain\user' format.

When Movicon is run by a domain user and a user group has been inserted in the project with the same name of the group that the user belongs to and who is present in operating system or domain, the user will be validated for the project management acquiring the access level set for that group.

When this group is not present in the project or the application is run by a local PC user, the user will be acknowledged by the local operating system or by the domain as anonymous at runtime with access level = 0 and no "CurrentRole".

## Shared Connection Repository

Used to generate one table for all the projects containing all the various project users so that they can be shared.

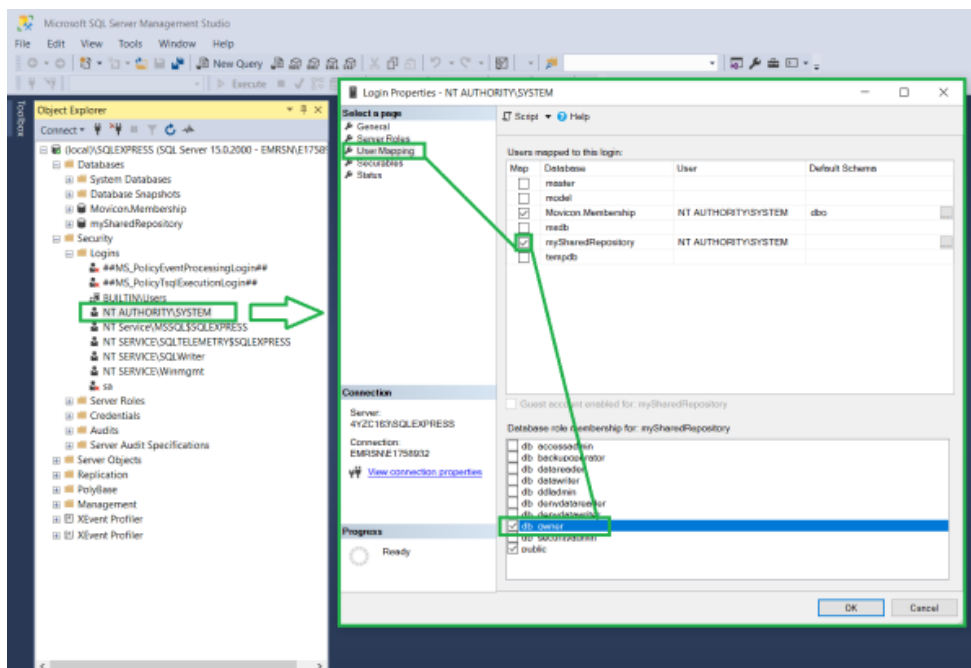


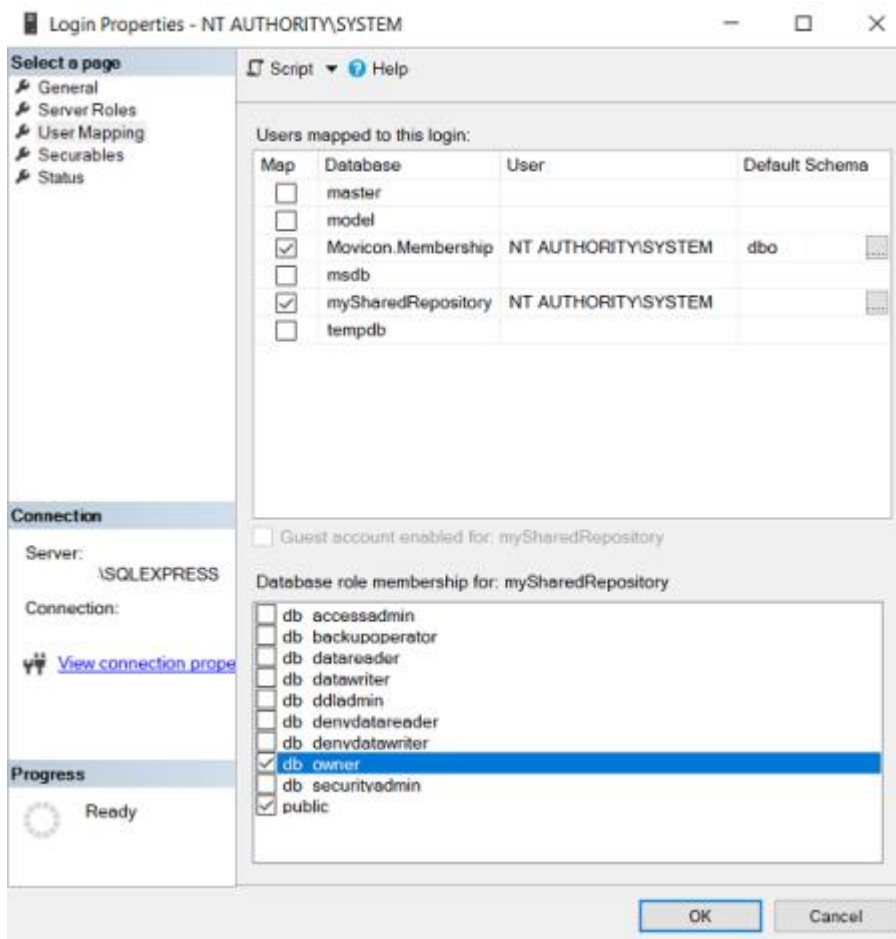
The Movicon.NExT installations used for each project need to have memberships in common. It will therefore be necessary to modify the "Moviconnext.exe.config" and Platformnext.exe.config" files so that they use the same "membership".

To share users between the Desktop Runtime project and WebHMI, you will need to activate the "db\_owner" membership role property of the user, who was selected during the "Movicon.NExT WebClient (SVG/HTML5)" service installation using the Service Control Panel, for the chosen Database for sharing users. If no specific user is indicated during the "Movicon.NExT WebClient (SVG/HTML5)" service installation, the service will be installed with the "NT AUTHORITY\SYSTEM" system user credentials.

To set the shared users Database access credentials (e.g. based on the SQL database engine) please proceed as follows:

1. Open the SQL Management Studio.
2. Select the "Security" folder and edit the user properties (you must add the "NT AUTHORITY\SYSTEM" user if not present).
3. Select the "User Mappings" item and activate the flag at the item corresponding to the Database chosen for user sharing.
4. Activate the "db\_owner" property and confirm with "OK".





## SMTP Property Settings

In addition to the general property settings, there is a **"SMTPSettings"** property group that allows you to set the report notification email functions.

**Attention, these settings do not concern the Alarm Dispatcher Notifications which need an appropriate Server configuration.**

### Server Address

Sets the address of the SMTP server address.

### Static from Address

If set, this allows the sender's address to be shown in the notification email to be generated by the system.

### Port Number

This parameter allows you to specify the port to use (default value 25).

Depending on the port number used, the SMTP client will use the appropriate security type. The supported protocols are: SSL, TLS and STARTTLS.

### Enable Authentication

Enabling this will activate the authentication on the SMTP server (when required).

### User Name

This field is used to set the user name to be used when SMTP Server authentication is required.

## Password

This is used to set the password to be used for the SMTP Server authentication when required.

### 13.4.2. User Management

In order to utilize the User Management in your project you need to create a list of Users or Groups with profiles. When you insert Users and/or Groups in the project you can give them access rights and privilege levels that are needed for executing project commands.

Even though the Membership technology is used with the SQL repository to manage user authentication, the Users and Groups have to be created using the Movicon.NExT project Users Editor. This is the only way possible to assign users with the specific information that is tied to the application that would not otherwise be assigned if using the Provider.



The Users or Groups Editor can be used in project edit mode, therefore while being programmed. It can also be used in runtime by authorized operators according to the preset modes and limits.

Users can be centralized in the **Domain of the Operating System** being used. In this case the Users Management has a profile managed by the Domain network of the Windows Operating System where the Movicon.NExT project is run. Therefore users can be managed with both modes: if a user does not exist in the project, the domain can be asked to authenticate the user's existence among those in the Domain users.



User authentication during runtime will be performed by the Provider by passing the credentials of the user, requesting the LogON, to the Movicon.NExT.

A Windows user can also be authenticated without needing to insert a Group. When no folder is created containing a group name in the 'Users' resource, Movicon will create three predefined group folders as **Admin**, **Guest** and **Power User** when the project reopens. If these folders are cancelled, they will be automatically recreated at the next project startup unless at least one group already exists in the resource.

In cases when the 'Enable Windows Authentication' option has been selected, Movicon will attempt to authenticate a Domain User and associate this user to the group created in the 'User and Group' (i.e. 'Domain Users').

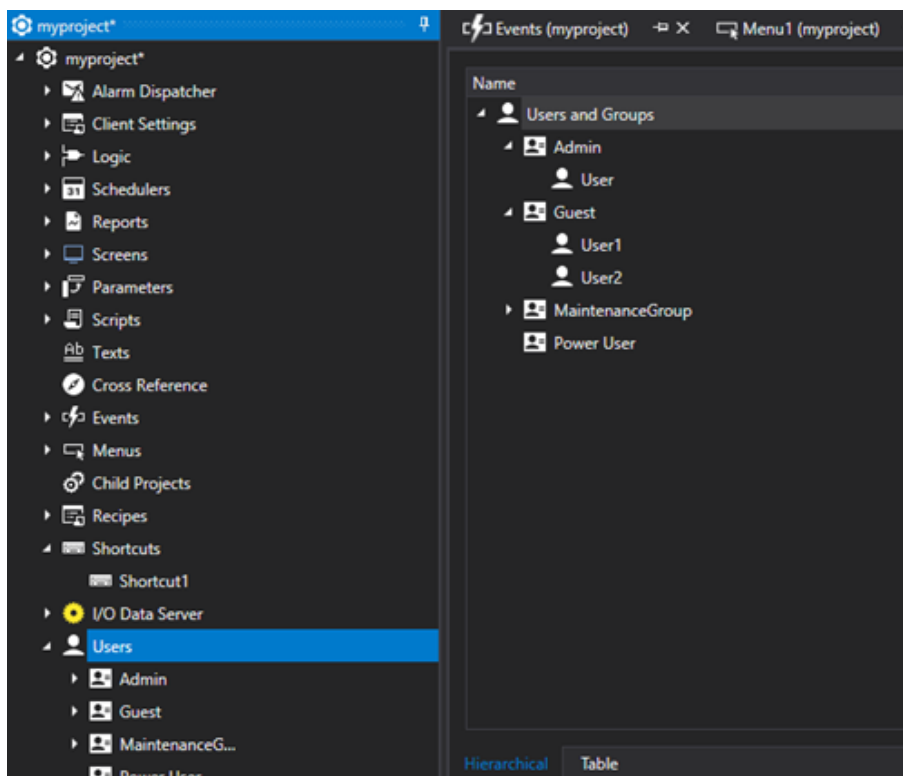
If the user is successfully authenticated, they will obtain access level rights of that same group.

If no user group exists, the user will obtain level 0. In cases where user groups exist without any domain users, the login will fail.

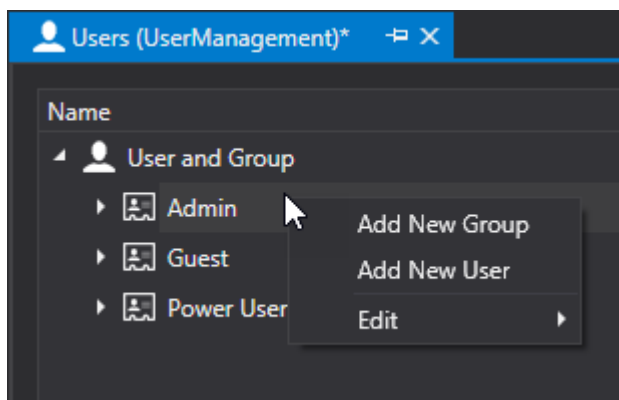
## User Management in the development environment

The project's Users management Editor can be accessed from the "**Users**" resource from the project's tree structure. This editor is used for setting the User Manager's general properties, or for entering and configuring users and their access rights to project commands.

The users that are defined in the project will also be created automatically in the Membership Provider's SQL repository at the startup of Movicon in runtime if not already existing in the Provider's list of users. In this case the Provider Membership user settings will not be overwritten by the properties set in the project for local users.



The commands for editing Users and Groups are made available after selecting the Project's "Users" resource from the Project > Users menu, or the contextual menu, within the development environment.



### Add New Group

This command allows you to add a new User Group to the project's user list. When clicking on Add New Group a popup window will appear containing the Group properties.

### Add New User

This command allows you to add a new User to the project's user list. The User will need to be added to a Group. When clicking on Add New User a popup window will appear containing the user properties.

### Delete

This command is used to delete the selected User or Group. **This command is only available in design mode.**

## Users Editor in Runtime

New users can be entered in the project during runtime as well , for example when new staff start while plant is already running a production process. The new users that are inserted in Runtime will be added to those previously inserted.

In order to insert users during runtime mode, the programmer should provide the necessary commands to display the Users Editor beforehand and as described in the topic on "User Manager Commands in Runtime".

### 13.4.3. User Properties

The User properties are used for managing the user's personal data that is needed for user authentication in order to access project functions and controls which have been activated with the Users management. Apart from the necessary access rights and privileges, user must also have Group membership.

Each user can be configured with the following settings using the properties window:

## General Properties

The General properties group are used to defined the main User settings.

### User Name

Name of User. The name must be unique to that user on the list and can contain a maximum of 64 characters.

### Password

This property is used to set a Password to associate to the User. The characters are hidden for reasons of privacy when entered. The minimum number of characters that can be used in the password can be set in the User Management's General Properties. The maximum number of characters allowed is 64.

- **All User password data are encrypted in the project therefore the right procedures must be taken to prevent passwords from being forgotten or lost!**

### Confirm Password

For security reasons, this field requires that the Password set in the above field be entered again and confirmed.

### Electronic Signature

The value of this property must be unique in order to identify the user clearly. If inserted, It's value will be recorded in a UserName column row in the relative Trace table (UFUAAuditDataItem). If no value is inserted, the user's name will be inserted in this column instead.

Electronic Signature uniqueness is handled by the system. In cases using Windows authentication, whereby no users Movicon.NExt project level are defined, Electronic Signatures are not defined and the user's name complete with dominion is reported in the Username column. To ensure Electronic Signature uniqueness, it is advisable not to have a mixed authentication model: Project User side and Dominion User side.

### User Language Activation

This property is used to associated a language to the user. If this field is left empty, the language defined in the Group of which the user is a member will be used. Therefore when the user logs in, the project will automatically activate with the language associate to the User in this field.

### User Converter

This property is used to define the Unit Converter associated to the User. When this field is left blank, the Unit Converter defined for the Group which the User belongs to will be inherited. In this case, when the user logs On, the Unit Converter associated to the user will automatically activate in the project.

### Password Expires In Days

The password associated to the user can be set with an expiry time in days for security reasons. When the set time expires so will the validity of the password. Therefore when the user next logs in after their password has expired, they will be requested to enter and renew their password with a different one. For reasons of regulation compliancy this procedure helps prevent the risk of the password being discovered and used by unknown parties.

The password will not be given an expiry time when this field is set at zero for default.

### Force Password Change at First LogOn

When enabled,, this property will oblige the user to change their password with a different one after the first login.

For reasons of regulation compliancy, this procedure prevents the risk of the password being discovered and used by other users and unknown parties when assigned by the Administrator.

## Access Level

### Access Level

This property is used to define the User's hierarchical access level. This level can be set with a value from 0 to 9999. The value set here will determine the user's level of access. The higher the value the more the user will have access to the project. For example, when a user is set with a level 10, they will be able to access and execute those commands that require user level equal to or lower than 10 (between 1 and 10). The user can inherit the Level associated to the Group he or she is a member of by entering the -1 value in this field.

In cases where user is authenticated by the operating system's Domain, they will receive the Group's hierarchical level.

### Access Area Mask

This property is used to define the user with an access area by means of using a bit mask. Each bit corresponds to an access area which allows the user to interact with those controls which require, in addition to the Access Level, the same Access Area. For example, a control can be set with an Access Level as well as enabled with Area 1. When a user logs on with the same hierarchical level or higher but has not been enabled with Area 1, they will not be able to use the control.

The Access Areas start from 1 to 31 which can be all selected together or individually as required through the Access Area Mask Editor popup window. As with Access Levels, the User can also inherit Access areas of the Group they belong to. This can be done by using the "Inherited" command button at the bottom of the Access mask Editor popup window. New users are created with inherited Access Areas for default.

## Notification

The properties in the Notification group are used to define the notification recipient addresses of each user. **These addresses are used for all notifications including those from the Alarm Dispatcher notification services.**

### Email - Address

This property is used for defining the User's email address. This address will be used by the Alarm Dispatcher to send the user notifications.

### Voip - Telephone Number

This property is used for defining the user's telephone number. This address will be used by the Alarm Dispatcher to send the user notifications. The number is displayed correctly, including the use of prefixes and national. International as provided by telephone operators.

### Voip- Tag Phone Number

This property allows to assign an I/ODaServer tag containing the user's telephone number. The assigned variable must be of type string and will contain the telephone number that the VOIP plugin will use to forward the notification.

### SMS - Telephone Number Mobile

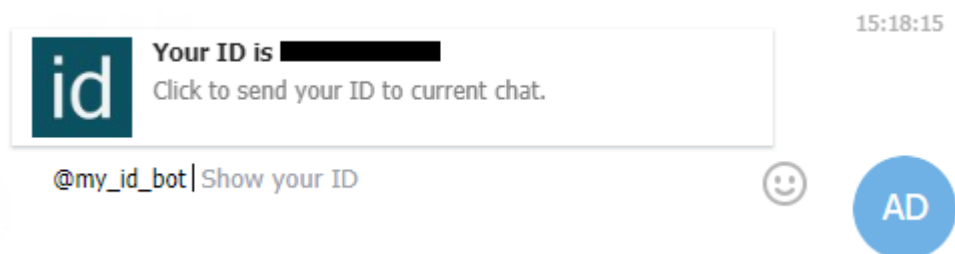
This property is used for defining the user's mobile phone number. This address will be used by the Alarm Dispatcher to send the user notifications. The number is displayed correctly, including the use of prefixes and national International as provided by telephone operators.

### Telegram - Chat ID

This property identifies the Chat ID of the Chat Telegram to which the Bot of the Alarm Dispatcher has been added to.

To retrieve the Chat ID proceed as follows:

- open the Telegram app
- start a chat with the Alarm Dispatcher's Bot (use the magnifier icon to search for it)
- select "Send" to start chat
- digit the "@my\_id\_bot" message
- the Chat ID to use in Movicon.NExT will show



The Telegram Chat ID property, when set, always has priority over the Telegram Group ID property.

### 13.4.4. User Group Properties

The User Groups are essential to the project's Users management. Users cannot be inserted individually unless they become members of a group first.

The Movicon.NExT User Management offers these default groups: Admin, Guest, Power User. These groups can be modified, removed and new ones can be added. When a "Group" is selected within the workspace, its property window can be opened to define the following settings:



### Name

Defines the Group's name. The group must have its own unique name within the group list which must be no more than 64 characters long.

### Default Access Level

This property is used to set the Group's hierarchical access level. This level will be used by those users who have not been set with an access level or who are authenticated by the operating system's domain. The hierarchical level can be set a value from 0 to 9999. The value set here will determine the user's access rights. The higher the value the more access rights the user will have. For example, a user with level 10 will be able to activate controls requiring a level equal to or lower than this value (comprised between 1 and 10).

The Level -1 cannot be used for Groups, but can be used by the users to inherit the Group's level.

### Default Access Mask

This property is used to define the access area using a bit mask for the Group. This mask will be inherited by those users who have not been set with one or who are authenticated by the operating system's domain.

Each bit corresponds to an access area where the user is permitted to interact with those controls that require an Access Area in addition to an Access Level. For example, When a control has been set with an Access Level and enabled with the Access Area 1, users who login with an Access level equal or higher than that of the command's but who do not have an Access Mask will not be able to use the control. The Access Areas start from 1 to 31. They can all be selected or only those needed by using the selection window. All Access Areas are enabled for default.

### Default Language

This property is used to define the language to be associated to the Group in order to automatically activate the language, if included in the project's Text Table resource, to convert all the project texts and strings to the language desired.

Users who log in without a culture name will automatically be associated the one set for the Group they belong to.

When no language is defined in this property, no text language change will take place.

### Predefined Convertor

This property is used to define the Unit Convertor associated to the Group. This Unit Convertor will be inherited by the Users belonging to the Group who have not been associated with their own Unit Convertor and which will activate when the User Logs On.

## 13.4.5. User Commands

The parameter settings of "Users" command types set to execute operative commands for Users in the Password and Protection management are configured in the following properties:

### CommandType

The command type to be executed.

- **Login:** executes user login. A dialog window opens for inserting the user's name and password.
- **Logout:** executes logoff of the user currently logged in.

- **UnlockUser:** A string type tag can be associated to the command with the name of the user to be re-enabled as its value. In cases where a tag is not specified or has a null value, a new popup window will open containing a list of users names that have been blocked within the project.
- **UnlockAllUsers:** This command unlocks all the users disabled within the project.
- **Edit:** Opens the users edit window through which it is possible to add or change users at runtime.
- **Change password:** Sets the command to change password using a pop-up that displays after activating the command.



To set the desired restrictions for the "edit" command, you will need to specify the "access level" or "access role" properties of the button used to invoke the command.

### Delay Command Secs

This command functions normally when set with the default value. When set to a value higher than 0, it only functions when kept pressed down for duration of the time set in seconds.

Commands activated in this mode (delay) exclude any normal commands eventually associated and vice versa.



This modality is not available in **webclient**; when a button is kept pressed down, all the delay commands that it has been associated with will be executed once only in sequence.

### 13.4.6. Domain User Authentication

When this functionality is enabled, the User Management system will permit access to users registered in the Domain of the Window OS in which the project is run, by using the login procedures. Therefore if the user wishing to access the project is not on the project user list, the user manager will ask the operating system for user authentication by means of using the LDAP protocol with the domain/User' format.

The user who launches Movicon.NExT (as service or as runtime) should belong to the Window Domain managed by the Active Directory (LDAP), otherwise this type of authentication will not function correctly. When the user is acknowledged, they will then be authenticated by the domain and therefore authorized to manage the project. In this case, the access rights will however be determined by the group the user belongs to.



Once authorized, the user's group membership is then validated and if a match to a project user group is found, the user will be granted access with the same access levels of that group.



The names of standard User Groups created by Windows (as 'Administrators' or 'Users') is NOT supported by the validation procedure. If you wish to use this authentication method, you will need to create new User Groups with custom names, then create the same user groups within the project for setting the access levels.

## 13.5. CFR21 Part11

### 13.5.1. General Concepts

#### CFR21 Part 11 General Concepts

The scope of the CFR21 Part 11 regulations issued by the Food & Drug Administration (FDA), is to obtain the legal equivalence of electronic documents (digital records and electronic signatures) to that of paper records and handwritten signatures executed on paper. This is due to the increasingly frequent use of automatic systems in managing production process systems subjected to FDA federal agency approval and revision. In order for the automation and control system to be in conformity with the CFR21 Part 11 norm, it is necessary that recorded data are traced back to the responsible operator (Electronic Signatures). Furthermore, specific precautions must be taken to prevent falsifications or tampering of data recorded electronically, or that allow easy identification in cases of inappropriate use whether intentional or accidental.

Many pharmaceutical companies wish to take advantage of the benefits derived from the use of electronic signatures. It is astonishing how much paper, that has to be stored, accumulates over the years. In addition, the use of electronic records significantly reduces the time needed to retrieve and revise these documents before releasing medicine for sale on the market.

These companies should apply for equipment that have the necessary mechanisms to protect against accidental or malicious modification of data in electronic format.



#### General concepts for supporting this regulation

The following concepts relating to the 21 CFR Part 11 regulation define how it would be advantageous to use Movicon.NEXT to develop FDA CFR Part 11-ready projects.

The basic concepts have been listed below to give better clarity on the understanding that the user takes full responsibility to ensure that their application developed with Movicon.NEXT conforms with the relevant requirements.



**Please remember that the hardware-software application is always to be validated in its entirety and not each individual product or component. It is the user's responsibility to create projects to conform to the 21 CFR Part 11 regulation.**

#### Management to Trace Variable Modifications

Tracing variable modifications will be managed by Movicon.NExT using the Historian (historicals and data logger), therefore you must make sure the relevant option is activated on the license.

If this option is not activated you will still have use of the Audit interface's functions without being able to execute any traces.

Different properties will be involved in order to manage the Audit Trace of variable modifications. For further information please click the link relating to the topic of interest:

## Objects

Data validation is performed using the 'Audit Trail Validator' control object from the Movicon.NExT toolbox.

The Data Source to be verified can be specified in the control's properties.

- Audit Trail Validator

## Users and Passwords

All the application commands that can be executed by operators to interact on the process must be protected by passwords.

The password management must be enabled in the project's User Password resource's General Properties.

- Electronic Signature
- Name (ID) and Password
- Enable Password Management
- Password Expires in Days
- Force password change after first Login
- Minimum Password length
- Auto LogOut timeout

## Tag

Movicon.NExT offers the possibility to trace all status variations and those of each variable value with significant relevance or that influence the production process. For example, an appropriate system to trace all significant process variables, such as set-points or process commands) should be ensured by using what is now as an Audit.

Movicon.NExT offers an Audit property group to allow users to define how the Audit is to be managed for each individual variable concerned.

- Max Audit Age
- Access Level Required to Confirm
- Enter Comment On Audit
- Enable Audit Trace
- Force Password On Audit

## Server

The audit-trail's role is to record all activities performed on variables by users during runtime. The recording of these activities is enabled by simply selecting the 'Enable Audit Trace' option in the variable properties. To validate a data set record in the database to ensure that it does not get altered externally, you will need to enable the 'Enable Data Protection' property found in the I/O Data Server's 'Settings' section. When this property is enabled, the I/O Server will startup with a certain user that is created by the Movicon Setup. This user, whose name for default is "NExT\_IO\_Server", and their password will be encrypted and will be used by the system to manage recordings in the Database. It is also the only user that can validate data by using by the "Validatore Audit Trail" object from the ToolBox.

- Default Audit Trace Connection
- Enable Data Protection in File



In certain procedures you will need to pay attention to the user with whom the Movicon.NExT I/O server is started up with. These procedures are:

- when deliberately changing the user in the Movicon.NExT server's 'Service Control Panel'.
- when installing the Movicon.NExT server's service using the 'Service Control Panel' before enabling the data protection option.
- when setting databases with a default connection string within which the authentication of a certain user is forced.

## Historian

- Enable Data File Protection



Setting the database with a custom connection string (DB Connection String property) and specifying a certain user to access the DB will prevent data from being validated.

## Datalogger

- Enable Data File Protection



Setting the database with a custom connection string (DB Connection String) and specifying a certain user to access the database will prevent data from being validated.



**Click CFR21 Part 11 to download the document.**

### 13.5.2. Security

To ensure project security please take the following points into account:

- The Movicon.NExT project must be encrypted so that all the configurations and passwords used in the project cannot be accessed from the outside.
- Movicon.NExT guarantees the uniqueness of users inserted in the project. Each user is identified in the project with UserID, Password, Description or a unique printable Name (electronic signature).
- The Movicon.NExT server side must be executed as Windows OS Service. By doing this access to the operating system and the recorded records will require identification of users registered in the project according to the security requisites required by the norm.
- Movicon.NExT supports shared Windows operation system domain in order to utilise user passwords defined by the system administrator.
- Users who manage data recordings using Data Loggers must adopt the appropriate measures to prevent unauthorized access to registered databases to avoid any possible tampering or undesired modification of their contents. When using ODBC storage, use secure databases such as the Microsoft SQL Server and administer the correct Windows OS security procedures permitting access to records to system administrators or developers only.
- To restrict access to the developed application's functions and controls, the Movicon.NExT project must use User Password Profiles management correctly by entering the Password, UserID, User Name and Access Level. Movicon.NExT provides 1024 access levels and 32 areas.
- Users must have passwords that can be managed with top security. The insertion of new users by the administrator can result in the subsequent password re-entry by the user when they next Log On.
- All the passwords can be set with expiry times to make users re-enter their password periodically to contribute towards increasing security.
- To correspond correctly to the norms, the Auto LogOff function (enabled access timeout) must be used correctly in the Movicon password management to avoid unauthorized access to the system after a period of user inactivity.
- To ensure correct and valid data entries, users must make sure that the Movicon.NExT operating stations are allocated in safe workstations that are only accessible by authorized personnel.
- The use of the Movicon.NExT Auto LogOff function is compulsory in systems that are in continuous use.

- Movicon.NExT adopts tools and procedures that the Windows OS uses to discourage the continuation of unauthorized access attempts as required by the norm. After the fourth failed attempt to Logon, Movicon.NExT will delay response time needed to re-enter the password to discourage attackers.
- System violation attempts. In the event of a fifth unauthorized LogOn attempt, Movicon.NExT will visualize and record the event in the Historical Log in order to control any further attempts of force entry into the system.

### Miscellaneous

- All data must be stored in a relational database that meets the security requirements (eg. Encrypted IMDB or ODBC with protected access against tampering and unauthorized data access by also using the security functions of the Windows OS). Data must be stored in a filing system for an adequate period of time according to operation needs.
- To safeguard data of the project, images and recipes even further, the user should use third party software products that can ensure version maintenance (for example, Microsoft Source Safe can be used to control versions).

## Electronic Signature

The Control Systems must be capable of acquiring the status and behaviour of the process's variables in real time. The date and the product batch number must be entered along with the electronic signature of the operator and an eventual signature of approval from the process manager in the section relating to the product batch's working period. These procedures must be carried out without the threat of causing errors and that signatures are always unique and referable to their owners. The records must be filed in a safe place and stored for an adequate time period. They must also be protected against unauthorized access.

### Security

Security in systems subjected to validation is absolutely essential. There are two cases in which data is recorded in electronic format.

1. Handwritten Signature: when data is printed and signed for approval (the so called Hybrid solution: paper and electronic). In this case the file is to be considered an electronic record. Adequate measures must be implemented to ensure that the file and its data contents are not substituted or modified before being printed, identified, dated and signed. When signed manually, it may not be necessary to use electronic signatures. Therefore, certain conditions must be setup to impede unauthorized manipulation of the data format and that such data is uniquely and automatically associated to a specific production batch or line. Furthermore, the original data file must be archived. It must also be stressed that a handwritten signature does not necessarily legitimize an electronic record inadequately protected.
2. Electronic Signatures: in this case everything is digital and all records are filed and stored in electronic format. In addition to ensuring that the file and its data contents cannot be substituted or modified, it must be also approved with an electronic signature. The data file should include information on the production batch it refers to and the name of the person who approved this data, being the person registered as logged on when data approval took place. All the file's original contents should then be protected from any unauthorized modification and manipulation.

### Electronic Signature

The electronic signature can be created with a combination of at least two items such as an ID code and a password or a badge and a password etc., as required by the CFR21 part 11. The User ID and Password must be guaranteed that it is unique to that person with absolute certainty of identifying them. The ID code can be made public, meaning that it can be shown on screen. Since the password may not always

be guaranteed as being unique to just one person, it is absolutely necessary that the ID code be original and personal to each user. These rules should be followed:

- A set minimum password length
- Change password periodically
- Carry out procedures to avoid any attempts of meddling or unauthorized access
- Record any attempts of unauthorized access
- The system administrator must not know the password of other users even when assisting them when they have forgotten their password.
- User Groups can share the same password only for reading data where the electronic signature is not required.

### 13.5.3. Validation and Documentation

The system used to validate data that has been submitted to the Audit Trail is based on a maximum security model that involves the use of encrypted system users and SQL Server Transaction Log verification. All data submitted to the Audit Trail are recorded according to univocal security criteria and can be validated by means of using graphical objects, called "Audit Trail Validator" available from the Movicon.NExT toolbox, that are used to view and timestamp data.

In order to get positive results, the validation process examines historical data to identify any tampering performed externally with Movicon. Each unauthorized variation will inevitably be detected by the Transaction Log analysis along with any user who is not a Movicon encrypted user ("NExT\_IO\_Server" user) but is responsible for performing the operations.

Please take the following points into account when dealing with the necessary validation and documentation concepts:

- Some of the CFR21 Part 11 regulation requirements demand activities and measures that are not based on the software application. To meet these demands required by the Part 11 standard, the customer must validate their application to ensure data recording accuracy, reliability and security in addition to its capacity to prevent the mishandling, errors and cancellation of data. Movicon.NExT users should validate applications developed in accordance with the FDA regulation standards. Users can develop and/or run their own validation programs or protocols or delegate this process to other entities. The validation process should follow a methodology established according to the system's life cycle (SLC).
- In order to meet the control procedures required to obtain conformity with the CFR21 Part 11 regulations, the customer must adopt adequate procedures to verify the identification of the individual assigned an electronic signature.
- The customer must establish in writing and put into practice the procedures to give specific operators specific responsibility for executing operations assigned to them according to their electronic signature to further impede falsifications or mishandling of their signatures or user registration in accordance with the CFR21 Part 11 regulations.
- The customer must verify the true identity of the individual to whom they wish to assign an electronic signature. In addition, the customer is required to certify in writing to the Federal Agency (FDA) that they intend to use the electronic signature as and equivalent substitute for tradition paper documents and, if necessary, produce the required documentation as requested by the agency,
- The customer is responsible for producing documentation on system use or the application they have developed and details of the produced documentation's updates and distribution as well as personnel training. However, the customer is not responsible for any documentation of the platform being used (Movicon, Windows).

- When producing 'guaranteed' documentation, the customer must use the timestamp viewer tools provided by the Movicon.NExT platform which can validate and guarantee the veracity of recorded historical data as appropriately predisposed in the project properties.

### Audit-Trail

The audit-trail records all operations carried out by users on process variables during runtime. The audit-trail is enabled to record by simply selecting the 'Enable Audit Trail' option found in each variable's Audit properties.

The 'Enable Data Protection' property, found in the I/O Data Server settings, protects both historical and Audit data so that they can be validated using the Audit Trail Validator object from the ToolBox.

When enabling this option, the ServerIO will startup with the "NExT\_IO\_Server" user created in the Movicon SetUp phase. This user, whose password is encrypted, will be used by the system to manage records on the DataBase and will be the only user permitted to validate data with the above mentioned object.

Below are listed some of the main columns or items that can be displayed within the "Audit Trail Validator" and "Historian Viewer" to keep track of eventual modifications to variables:

- **Name:** indicates variable name.
- **Description:** this is the description associated to the variable using the Tag's 'Description' property.
- **Value:** indicates the Tag's value after being changed.
- **ValueBefore:** indicates the tag's value before being changed.
- **Status:** indicates the Tag's quality.
- **RecordDateTime:** indicates the date and time of the event.
- **UserName:** indicates the name of the user who generated the event.
- **Reason:** Comment entered by the user while making change (This requires that the 'Force Comment On Audit' property be enabled beforehand).

### Data Validation

Data validation is done by using the 'Audit Trial Validator' object from the Movicon.NExT toolbox.

Data validation in Movicon.NExT is essentially based on the SQL Server's Transaction Log. Basically, the logs of protected databases are monitored to see if any record sets have been modified by unauthorized users who are not registered with the Movicon.NExT I/O Server ("NExT\_IO\_Server" user).



When the project is started up for the first time, the first database backup is performed automatically after the first data recording in the SQL Server database has been executed (SQL Server installation backup folder). This first backup is **essential** for subsequent data validations. If this first backup is removed or deleted, it will not be possible to validate data within the database.

In addition, the validation of data in the Transaction Log is not based on the actual validation user's name (default "NExT\_IO\_Server"), but on their SID (the user's security ID). If the validation user (default "NExT\_IO\_Server") is eliminated by the operating system and then reinstated, their SID will change and therefore it will not be possible to validate data recorded with the previous user.

### Time Stamp (date and time) Management

- The Time Stamp is managed by Movicon.NExT using the Windows operating system clock for both date and local time which is Universal Time Coordinated (UTC).
- In order to get the right time, the user should set the operating system to synchronize with the metrological servers that refer to the Network Time Protocol (NTP), or synchronize the client system date and time with the Server's for data recording consistency. These synchronizations can be managed directly using the Windows 7, Windows 8 and Windows 10 operating system functions or by using Basic Script code to synchronize the project times.



### Validation User (NExT\_IO\_Server)

The encrypted and unique validation user ("NExT\_IO\_Server"), which is essential for the validation of Audit Trail data, is added to the operating system during the **Movicon Setup** phase. This user will be added as local user to the machine with the name of "NExT\_IO\_Server" when executing a Standard Setup. When executing a Custom Setup, a different user name can be specified or a Domain user can be created/selected if the PC belongs to a Domain.



The Domain user can only be created by accessing with a Domain user who has the necessary credentials for executing this operation.

The validation user ("NExT\_IO\_Server") will also be added to those of the SQL Server and added to the list of users authorized to start the Windows services.



If the validation user (default "NExT\_IO\_Server") is added as domain user, the Movicon client application should also be started up with a domain user that has the right credentials to retrieve the validation user's information (default "NExT\_IO\_Server").



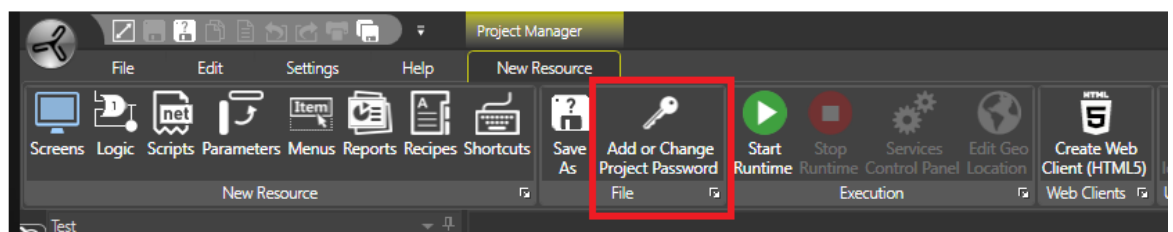
When the Movicon.NExT I/O DataServer is run as service, the validation user (default "NExT\_IO\_Server") should be the same one with which the service was started up, unless the user is deliberately changed in the 'Services Control Panel' or the service is installed before enabling one of the data protection options.

### 13.5.4. CFR21 part 11 configuration

To get a Movicon.NExT project 21CFR Part 11 ready, you will need to configure it appropriately so that it is compatible with the FDA validation criteria. The necessary measures to take in doing this are indicated below:

#### Project Security

The project must be configured in its General Properties by selecting Project Password and setting a protection password. In this case, the project access in design mode will be protected and all the project's XML will be encrypted and inaccessible due to a proprietary algorithm.

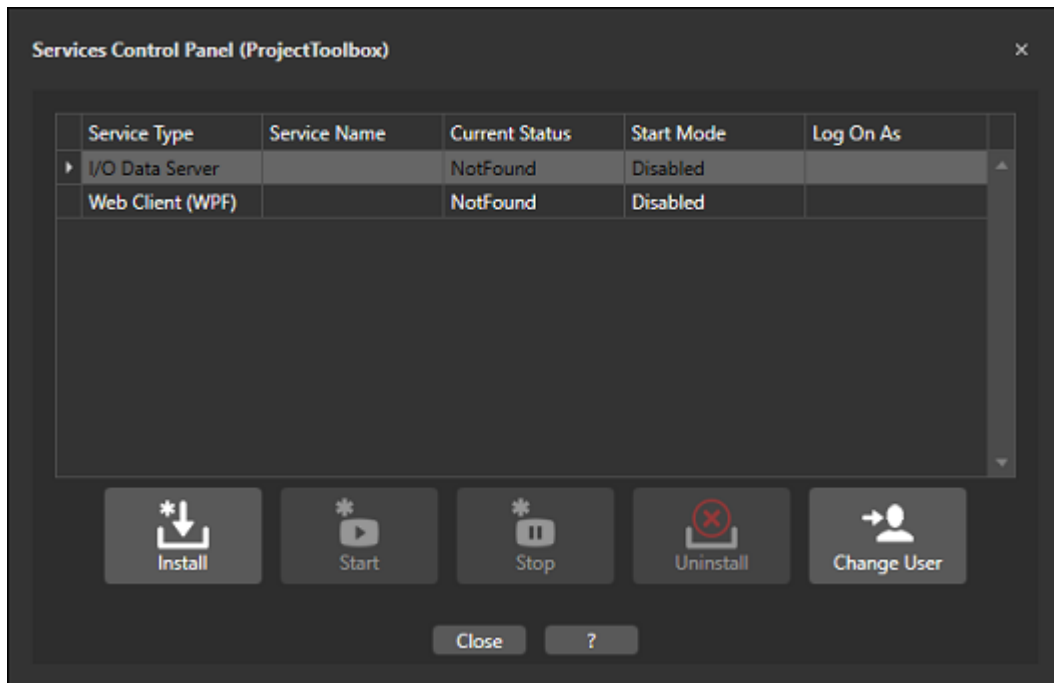


Confirming this operation will save all the Movicon.NExT project files with encryption and protection when the project is next opened. It is important to remember that projects protected with passwords can be executed in runtime but cannot be modified or accessed in design mode without using a password to gain access.

#### System Startup

- When turning on the PC, the Windows operating system will startup together with the Movicon.NExT platform by automatically launching the project. For security purposes, these operations should be configured appropriately to ensure that the project is started up safely in protection mode.

- At Windows startup, all the Movicon.NExT project Server components should be run automatically using the Windows Services' functions. The Movicon.NExT project should therefore provide Services management using a control panel that can be accessed from the Movicon.NExT Editor. This Editor is used to install the Server parts (such as the I/O Server, Schedulers, Recipes and other) as Windows Services. In this way, they will all be run automatically when starting up the operating system independently from the Windows User Log On.



*The Services Control Panel accessed by means of the Movicon.NExT Editor*

- The Windows' user interface can only be accessed when Windows user has been authenticated. As a consequence, the Movicon.NExT user interface, known as the Client, will only be available after the Windows' User has Logged on successfully. There are several options to use for doing this:
  - Windows startup in normal mode by means of Windows user log on. The desktop interface will not be available until a Windows user has logged on. After a user has logged on with Windows, the startup applications can be run and then a command line can be executed to startup the Movicon.NExT Client. The Movicon.NExT Client project will then manage security and the project user management as well as consenting access to the Windows Desktop.
    - The Windows startup can be put into operation by setting the operating system to automatically log on a Windows user for default by means of using the operating system's startup command string. In this case, Windows will startup with an enabled 'standard' user, and as a consequence so will the Movicon.NExT Client startup automatically. The Movicon.NExT Client project will then manage the security and project users management as well as consent access to the Windows Desktop.
    - The Windows and Movicon.NExT user management can be shared. Movicon.NExT permits shared Windows User Domain, and as a consequence the Windows user Log On will permit Windows authentication and the startup of the Movicon.NExT Client project. In this case, the Movicon.NExT project users will be shared with those of the Windows operating system domain.

- Windows can also startup without the Desktop (including the resource explorer and management), and therefore by starting up the Movicon.NExT Client only by impeding any access to the Windows Desktop.
- In any case, access to the Windows desktop or its exclusion, and the configuration of Windows user privileges does not depend on Movicon.NExT, and must be managed by the system administrator.

### Access to the Windows user interface (Desktop)

While in operation, therefore after the startup of Windows and the Movicon.NExT project application, it is important to manage the operating system's functions that consent access to the desktop from the application in the right way.

According to which mode the operating system was started up in, the user has the option to manage controlled Desktop access by means of the access level settings associable to the project user passwords.

- When the Movicon.NExT user interface (Client) is running, unauthorized access to the Windows user interface (Desktop) can be blocked by eliciting Movicon.NExT user authentication request for users with Desktop access privileges. User role requests can be assigned to users with Desktop access by means of using the Movicon.NExT General User Management properties.
- The operation system can also be configured to exclude the Windows Desktop in order to use the Movicon.NExT user interface functions only. This configuration must be done by a system administrator.

### Passwords

All the application commands that can be executed by operators to interact on the process must be protected by passwords.

The password management must be enabled in the project's User Password resource's General Properties.

- Enable Password Management: the passwords will be activated according to the levels and access modalities to the preset commands.
- Auto Logoff: determines the time (sec.) for automatically deactivating the active user after a period of inactivity.
- Minimum Password length: sets the minimum password length allowed (4 characters for default).
- Electronic Signature Definition: the unique user Description of the user whose name is to be used as an Electronic Signature will be managed.

Movicon.NExT will automatically control correct authentications, the uniqueness of the Electronic Signatures and any attempts to force User Log On. Any further Log on attempts made after the fifth one fails, will be recorded in the Event Log and the response time will deliberately take longer to discourage further attempts.



The number of login attempts can be set in the Membership section found in the "MoviconNExT.exe.config" system file as shown below.

```

<membership defaultProvider="AspNetSqlMembershipProvider">
  <providers>
    <remove name="AspNetSqlMembershipProvider" />
    <add name="AspNetSqlMembershipProvider" type="System.Web.Security.SqlMembershipProvider, System.Web,
      connectionStringName="LocalSqlServer"
      enablePasswordRetrieval="false"
      enablePasswordReset="true"
      requiresQuestionAndAnswer="false"
      applicationName="/"
      requiresUniqueEmail="false"
      passwordFormat="Hashed"
      maxInvalidPasswordAttempts="5"
      minRequiredPasswordLength="1"
      minRequiredNonalphanumericCharacters="0"
      passwordAttemptWindow="10" passwordStrengthRegularExpression="" />
  </providers>
</membership>

```

Each user or user group, who have access to commands or process interaction, must be inserted and configured in the project appropriately.

Users are inserted in the project's User Password Resource where they can be configured in their properties. These properties include those which conform to the FDA regulatory requirements:

- Name (ID) and Password. These are assigned to the user and are used for identification by the system.
- This is a unique text which corresponds to the user's electronic signature and is recorded as absolute user identification (the Electronic Signature management must be enabled in the User Password Resource)
- Auto Log Off: user deactivation time to be used after a period of inactivity.
- Expiring Password: The act stipulates that the user password expires after a certain preset time so that the user is obliged to change it periodically to increase system security.
- Change Password after first login: the normative requires that user be made to change their password once Logging in to prevent the administrator who inserted the original password from knowing it and thus contributing more to identification certainty.

### Comments on Alarm Acknowledgements (Audit Trail)

In many cases, before the operator is allowed to acknowledge an alarm, they may be required to make a comment, which will then be recorded in the historical together with the alarm ACK event.

Movicon.NExT provides a function where a severity level for each individual alarm is specified and in addition to which the Movicon.NExT Alarm window will ask the user to enter a comment on the acknowledgement operation.

Therefore, it will be necessary to define the alarm's severity level and threshold level that once reached will ask the operator user to enter a comment on the operation performed.

1. A severity level must be defined in the properties of each alarm.
2. The priority threshold level must be defined the Alarm Window properties that once exceeded will force the operator to enter a comment on the acknowledgement operation for Auditing purposes.

Comments will be recorded in the Events log with the acknowledgement operation.

An User Access Level can also be set for executing operations in alarms.

### Data Storage Security and Validity

Protecting the storage of recorded data against manipulation of Electronic Records is absolutely essential to obtain validation and conformity with the CFR21 regulations and their application.

Data recorded by Movicon.NExT (Audit, Historian, Data Loggers) are physically created with database file archives. In order to ensure an effective data validation system, Movicon requires the use of the Microsoft

SQL Server as a database for recording data. Projects can use other types of databases for storing data but if you need to have a FDA CFR21 Part 11 compliant, the Movicon.NExT stored data integrity validation system is based on secure control mechanisms that require the exclusive use of the Microsoft SQL Server. It is therefore absolutely crucial that you comply with the system configurations and project design constraints described in the product documentation. Succinctly, Movicon.NExT validation system provides a mechanism to record data on databases using a data uniqueness criterion based on the encrypted Movicon.NExT User system and specific controls from the SQL Server system's Transaction Log. Therefore, Movicon will record data using a unique and encrypted Windows System User in association with the database's Transaction Log to control and detect any undesired manipulations. This pre-validated database containing information on the process and each Logged transaction will safeguard stored files submitted for validation from any undesired manipulation or modification. This ensures that the file is analyzed in its original form and validated through a Viewer designed to detect and alert any manipulation to data upon which validation of such file will be withheld. In addition to the above mentioned content, it is the responsibility of the user to implement the necessary data security and integrity criteria by configuring the database with the most adequate data access protection and retention systems that can ensure data availability for an appropriate period of time with reliable redundancy and backup systems as stipulated by the regulations.

### Electronic Records

By Electronic Records we mean all production process information (data, values, events) electronically recorded in archives that ensure data integrity and protection against undesired manipulation. All the information that Movicon.NExT records in database archives can be defined as 'Electronic Records'. In order for the Movicon.NExT Electronic Records to be compliant with the regulations, you will need to follow the indications and guide lines contained in this document and in the user manual to ensure data integrity and security and prevent unauthorized data access or manipulation.

Data recorded on the SQL Server database within the constraints of project design and application and with the relative intrinsic security criteria, are submittable for validation and authentication in order to ensure the originality of data and therefore to impede any type of unwanted manipulation. Movicon.NExT manages a recorded data validation system that can be activated by means of using the historical data management property settings configuration. This is done by checking the Enable Event Data Protection property in the Database Settings properties of the recording engines managed by the Movicon.NExT Server.

When enabling this function, Movicon.NExT will use an encrypted System User ID and Transaction Log combination to ensure the integrity of each individual data recorded (records) on DB.

Aided by this control, Movicon.NExT can ensure users that the data recorded is genuine and impede any possible manipulation to alter or cancel the data.



Each historical data storage system can therefore be subjected to validation controls in Movicon.NExT, to confirm data authenticity or detect any data tampering.

Users operating in systems subjected to CFR21 validation, will be able to ensure data authenticity by implementing it where needed using data protection management or data visualization or print validation checks.

When enabling the data protection management, it will be impossible for anyone to tamper with historical data recorded by Movicon.NExT as validation checks are run to validate data authenticity or highlight any alterations due to unauthorized tampering.

OID	Name	Value	dValue	ValueBefore	dValueBefore	StatusCode	Status	RecordDateTi...	RecordDateTi...	RecordDateTi...	SourceTi...
1	Tags.TagAudit.Tag1	1	3	3	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:519			30/08/2018
2	Tags.TagAudit.Tag2	2	4	4	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:891			30/08/2018
3	Tags.TagAudit.Tag3	3	5	5	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:257			30/08/2018
4	Tags.TagAudit.Tag2	2	1	1	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:537			30/08/2018
5	Tags.TagAudit.Tag3	3	2	2	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:195			30/08/2018
6	Tags.TagAudit.Tag4	4	3	3	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:667			30/08/2018
7	Tags.TagAudit.Tag3	3	2	2	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:660			30/08/2018
8	Tags.TagAudit.Tag4	4	3	3	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:220			30/08/2018
9	Tags.TagAudit.Tag5	5	4	4	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:771			30/08/2018
10	Tags.TagAudit.Tag4	4	3	3	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:307			30/08/2018
11	Tags.TagAudit.Tag5	5	4	4	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:837			30/08/2018
12	Tags.TagAudit.Tag5	5	4	4	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:380			30/08/2018
13	Tags.TagAudit.Tag6	6	5	5	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:876			30/08/2018
14	Tags.TagAudit.Tag6	6	5	5	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:419			30/08/2018
15	Tags.TagAudit.Tag7	7	6	6	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:634			30/08/2018

This screen shot shows Historical database subjected to validation with positive results. Instead, the screen shot below shows the same check on a tampered and unauthentic historical database.

OID	Name	Value	dValue	ValueBefore	dValueBefore	StatusCode	Status	RecordDateTi...	RecordDateTi...	RecordDateTi...	SourceTi...
1	Tags.TagAudit.Tag1	1	3	3	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:519			30/08/2018
2	Tags.TagAudit.Tag2	2	4	4	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:891			30/08/2018
3	Tags.TagAudit.Tag6	3	5	5	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:257			30/08/2018
4	Tags.TagAudit.Tag2	2	1	1	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:537			30/08/2018
5	Tags.TagAudit.Tag7	3	2	2	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:195			30/08/2018
6	Tags.TagAudit.Tag4	4	3	3	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:667			30/08/2018
7	Tags.TagAudit.Tag6	3	2	2	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:660			30/08/2018
8	Tags.TagAudit.Tag4	4	3	3	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:220			30/08/2018
9	Tags.TagAudit.Tag5	5	4	4	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:771			30/08/2018
10	Tags.TagAudit.Tag4	4	3	3	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:307			30/08/2018
11	Tags.TagAudit.Tag5	5	4	4	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:837			30/08/2018
12	Tags.TagAudit.Tag5	5	4	4	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:380			30/08/2018
13	Tags.TagAudit.Tag6	6	5	5	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:876			30/08/2018
14	Tags.TagAudit.Tag6	6	5	5	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:419			30/08/2018
15	Tags.TagAudit.Tag7	7	6	6	0	Good	Good	30/08/2018 15:11:30/08/2018 17:11:634			30/08/2018

## Data Retention

Historical data must be made available in storage for an adequate period of time according to the process managed. The period of time can be defined as needed in the Movicon.NExT recording engine and independently from the time of databased being used. Object properties, in effect, allow the number of days (e.g. 730 days can be set to cover 2 years) for which the application will ensure data availability. Once this period has been reached, the oldest data will be overwritten by the most recent to ensure that data is retained for availability for the period of time desired.

Data backup can be managed as desired in the project using the script functions to manage the data sources and destination backup files, or third party backup tools can also be used instead.

### Redundancy

Movicon.NExT totally supports the multi-server redundancy function, not just for archive synchronization but also for any operating functionality automatically and transparently. The Redundancy function should be applied and managed in a way that complies to the regulations and in function with the process type. The Redundancy function allows Movicon.NExT to synchronize the historical archives in a number of servers to ensure both maximum data and operation reliability.

### External Security for DB archives

By means of using the Movicon.NExT Server functions, process data are recorded on the configured database. These data therefore physically reside in files and tables that can be recorded locally on hard disk or in mass archives that physically reside in several servers, or in the Cloud. With the aid of safe relational databases, such as the SQL Server, Movicon.NExT uses protected connections (accounts) for accessing files. It is the responsibility of the user to configure the system so that no one can access files, by removing access rights to files both in the actual database and in the access privileges to folders by the operating system (Movicon.NExT run as service).

It is necessary to ensure data security by using this procedure:

- Movicon.NExT supports any type of relational database for storing data. However, to ensure that the right validation mechanisms are used, you must use the Ms SQL Server.
- To avoid unauthorized access to files, User Account protection will need to be setup by using the access criteria explicitly for system administrators or program designers only (eg. With the same project protection password). This will impede access to data tables where authorization has not been provided.
- Use the operating system's access lock (Locked by Movicon.NExT) or access rights to operating system by using Movicon.NExT as Service. By doing this, file access through the operating system will be physically denied.
- Do not share folders or disks when the station is operating in net, except for system administrator access.
- Remove all rights to modify database records (Updates). Movicon.NExT lets new records to be inserted whose data cannot be accessed for altering no matter what the reason is.

### 13.5.5. Data Backup Validation

It is very important that any restoration of backed-up data be done using procedures that ensure their integrity otherwise, even though original, it will not be possible to validate them.

### Data Backup

The CFR21 Part 11 norm explicitly stipulates that data restored from backup involve procedures that ensure their integrity for validation purposes. This requires that data subject to validation be backed-up and restored together with data of the Windows operating system user.

The inclusion of the Windows operating system user data is vital for validating data with the Movicon Audit Trail Validator which is based on the security identifiers (SID) of the operating system user contained in the backup data.

Data will not be validated if the process data is restored without the operating system user data.

Please refer to the Microsoft documentation on operating system image back procedures to consider when backing up data.





The validation of data is based on the SQL Server's TRANSACTION LOG using SID (user security ID) unique to each validation user ("NExT\_IO\_Server" User). This means that if the validation user ("NExT\_IO\_Server") is cancelled from the operating system and regenerated later, the SID will change. This is done to ensure that maximum security is maintained when validating data. Therefore, when the SID is changed, it will not be possible to validate data recorded by a previous user !

Therefore, in addition to backing up data contained in the SQL Server database (which includes the Transaction Log), you will also need to configure operating system backup in the part regarding Windows Users.

For more information on using the best backup strategy, please refer to the Microsoft SQL Server and Windows backup and restoration documentation.

### Data Restoration

In the even of a "Disaster Recovery", it will be necessary to proceed with restoring the previously created operating system's image.

Please refer to the Microsoft documentation for Restoration procedure of system image.

As regards to restoring the database, it is **essential** to restore backups based on the order in which they were created. Before restoring a specific transition log backup, you will need to restore all the previous backups that took place without executing a transition rollback by using the SSMS's 'WITH NO RECOVERY' option within the database restore window.

Please refer to the Microsoft documentation for the SQL Server database backup restore procedure.

### Transparent Data Encryption

The Transparent Data Encryption (TDE) is used in SQL Server, Azure SQL Database and Azure Synapse Analytics (SQL DW) data files known as encrypting data at rest. Several precautions can be taken to help protect the database such as designing a secure system, encrypting confidential resources and building a firewall around the database Servers. However, in a scenario where the physical media (such as hard backup drives or unit) are stolen, a malicious party can simply restore or attach to the database and browse through the data. One solution to prevent this from happening is to encrypt sensitive data in the database and protect the keys that are used to encrypt the data with a certificate. This will prevent those without the keys from using the data. However, this type of protection will need to be planned in advance.

TDE performs real-time I/O encryption decryption of data and log files.

The encryption uses a database encryption key (DEK), which is stored in the database boot record for availability during recovery. The DEK is a symmetric key secured by using a certificate stored in the master database of the server or an asymmetric key protected by an EKM module.

TDE protects data "at rest", meaning the data and log files. It provides the ability to comply with many laws, regulations, and guidelines established in various industries. The enables software developers to encrypt data by AES and 3DES encryption algorithms without changing existing applications.



Backup files of databases that have TDE enabled are also encrypted using the database encryption key. As a result, when you restore these backups, the certificate protecting the database encryption key must be available. This means that in addition to backing up the database, you have to make sure that you maintain backups of the server certificates to prevent data loss. Data loss will result if the certificate is no longer available. For information, see SQL Server Certificates and Asymmetric Keys.



### 13.5.6. User Sharing

#### Windows User Sharing

Movicon.NExT offers the possibility to share users from the operating system's Domain or that of Windows (Win7/Win10).

This will enable security system managers to use one defined point of network users by using the Windows Domain Users in alternative or together with the users defined in the project. Movicon.NExT accepts mixed configurations of users inserted in the project list and user from the Win7/Win10 domains, whose log on and management is delegated to the operating system. When using Windows users, the electronic signature field in the electronic records will assume the value given by the name of the user qualified by the domain they belong to: i.e. Domain Name/User Name. Mixed configurations do not ensure the uniqueness of information inserted in the Electronic Signature field due to the fact that Movicon.NExT does not have control over information relating to domain users. Therefore, we suggest you avoid using mixed configurations even if technically permissible.

When authentication of a User is requested in Movicon.NExT by means of the Log On window, the user will first be verified with those of the project, and if the 'Share Windows Users' property has been activated, Movicon.NExT will ask the operating system to authenticate the User Name and Password to access the operating system's Domain. The criteria that establishes the project's Access Level is defined in the User Group, as described below:

- The users defined in the Windows Domain (Primary Domain Controller) can also be authenticated in Movicon.NExT and be given a custom user level by means of using a common User Group definition. The Windows Domain Users need to be inserted in the Windows User Group whose name is identical and present in the Movicon.NExT User Group. Each Movicon.NExT User Group can be defined with an Access Level and Access Area in its properties. Therefore, when a User is authenticated by Windows, Movicon.NExT will receive authorization and assign the User Level defined in the name of the Group to which the user belongs. It is, therefore, possible to assign the desired password level and access area to each Domain User by means of using the Groups.

For example:

- Windows has defined the 'R\_Waters' user in the 'Machine\_Operators' group.
- The Movicon.NExT project also contains a User Group with the identical "Machine\_Operators" name.
- The desired Access Level and Area properties are assigned to users of the Movicon.NExT Group including "R\_Waters".
- The project should also be enabled with the Share Windows Domain Users property.
- When the user Logs On to Movicon.NExT, their access credentials will be passed over to Windows for authorization to then return back to Movicon.NExT positive. Movicon.NExT will then grant access to the user assigning them the access level defined in the Group.
- The same operation can be performed for any other user by sharing Groups or creating new ones as needed.
- This mechanism is also valid for users configured directly during runtime with the Movicon.NExT user editor window.
- The user's Electronic Signature and Name (User ID) that Windows manages uniquely, preceded by the name of the Domain.

#### Biometrics Systems

Using Biometrics Systems is highly recommended in application validity according to the regulations. In this case, you need to choose the right recognition system among those available on the market that can be easily integrated into your application.

The most popular biometrics systems are ultimately those that use digital fingerprints. These systems are simple to use and integrate perfectly with operating systems and software applications.

### 13.5.7. Trace/Audit for Project User Management

User management operations in project runtime are traced in the project's log file's "UFUAAuditLogItem" table.

The currently traced operations are Log-in, log-out, and user editing at runtime such as:

#### Operations carried out in User Groups for:

- Creating Groups (Groups along with user contents can also be Copied/Pasted)
- Eliminating Groups
- Editing Group properties (Name, Access Level, Access Area, Language)

#### Operations involving Users for:

- User Log-in
- User Log-out
- User password change at first time log-in
- User password change for expired passwords
- User password change prompted by command
- Creating new Users (users can be copied/pasted into different Groups)
- Eliminating Users
- Editing User properties (Name, Password, Electronic Signature, Password Expiry in days, Change Password at first log-in request, Email, telephone no., mobile no., Telegram Chat ID. Access Level and language)

The data will be recorded in the "UFUAAuditLogItem" table and the following information for each audit event listed above will be entered in the table:

- **EventDateTime**: event's date and time of execution
- **SourceName**: hostname of the Client who executed the operation (Client1, Client2, ecc.)
- **EventMessage**: description of the event type executed ("Utente1" Log-in, "Utente1" Log-out, added "Utente2", "Utente3" property changed, etc.)
- **EventDetails**: additional event information (Password changed at first log-in, "Email" property changed, etc.)
- **UserName**: name of user logged in project when operation was performed.

# 14. WebClient

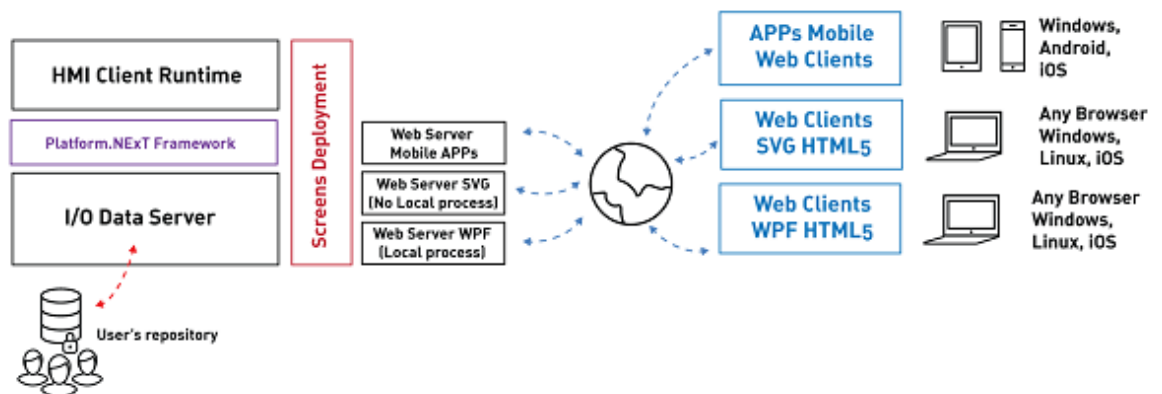
## 14.1. Introduction to using the WebClient

The Movicon.NExT Web Client technology is the most modern and innovative solution to use for managing remote access to the supervision system from any remote device or mobile by means of using local and public networks, VPN or internet with the maximum of simplicity and security.

The use of the Movicon.NExT Web Client is optional and must be enabled only on the Server's runtime license in order to allow access to the desired number of concurrent HTML5 (Browser) or WPF (Apps) users.

The diagram below illustrates an example of the Movicon.NExT web architecture. As you can see, the Data Server uses a Web Server application (IIS for HTML5 or integrated for WPF) and the screens, that have been created, can be published on HTML5 web pages or by means of an App using the relevant Editor command.

In this way users can independently access the Web pages and interact with the system by displaying screens. User login can also be implemented by configuring users and their passwords in the data server.

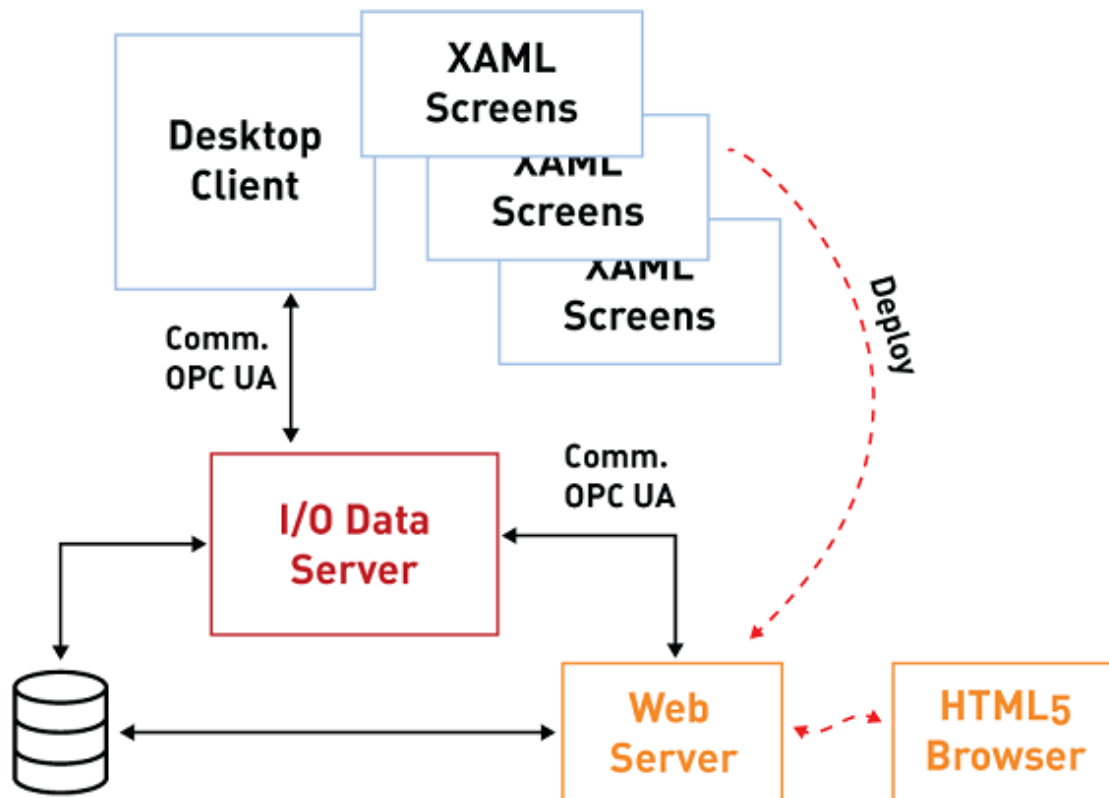


The Movicon.NExT technology enables you to use the Platform's Web Server features to deploy the supervision project on the Web and therefore to be accessed by remote Client devices using the HTML5 technology. Thanks to this new standard, Clients can access the server adopting the "Cross Platform" concept whereby the client can access the server using any browser and device with any operating system without having to install extra locally to enable this..

## 14.2. WebClient (WPF/HTML5)

### 14.2.1. Web Server Architecture

The system's architecture provides the use of the Microsoft Internet Information Services (IIS) Web Server that on one side manages communications with the I/O Data Server through OPC UA, and on the other side it exposes HTML5 web paged for Browser access.



Therefore as with the Movicon.NExT graphical interface for desktop clients, the Web Server acts as an OPC UA client towards the data Server.

**i** This Web architecture is totally independent. Therefore it is possible to create projects that have both a local Client HMI interface and a HTML5 Web Client interface. Otherwise a "blind" server can be implemented to publish graphical pages only for remote Web Clients.

Access to the Supervisor is performed on the Client side using a common internet browser (Internet Explorer, Chrome, Firefox, Safari, etc.), with different devices such as PC, Tablet and Smartphone. the HTML5 Web technology has the advantage of being Cross Platform-based because it is practically available in all the most popular operating systems (Windows, Windows Phone, iOS, Android, Linux, etc.). Thanks to the HTML5 technology, clients are no longer passive and get to manage bidirectional communications, user protection, database access and report presentations, within a Web context without needing any installation or configuration on remote client side.

**!** Even though there are many advantages of having distributed web technology, there are also some limits to its use on the HTML5 Client side as described in this guide. The user should consider these limits when project engineering by evaluating what of the two architectures, Desktop Client or Web Client , is most appropriate to use according to their needs.

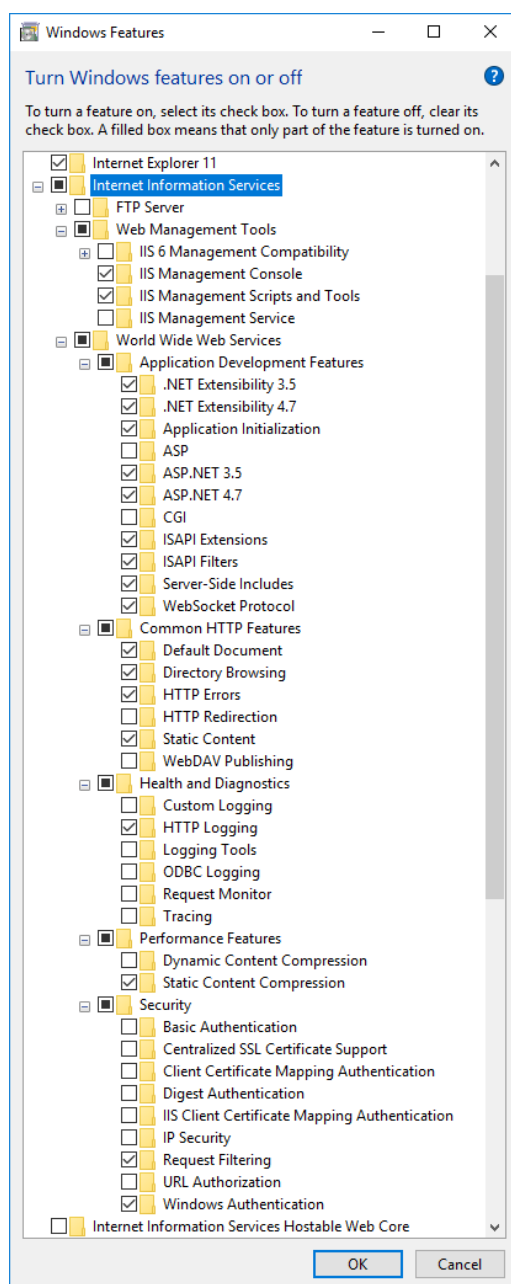
### 14.2.2. IIS Web Server Configuration

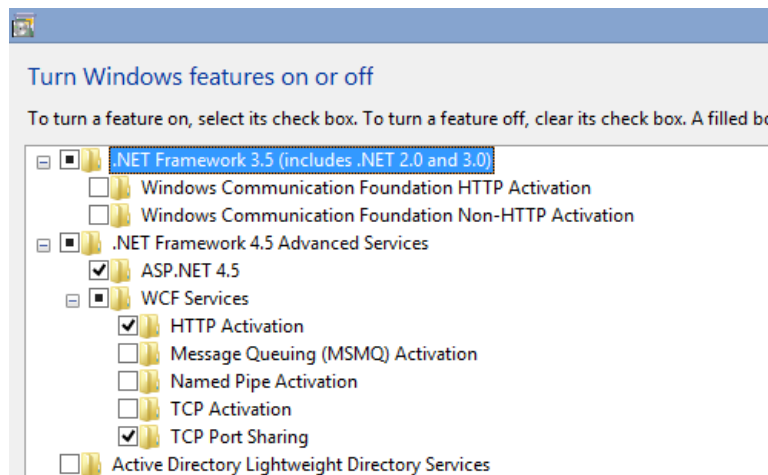
The use of the WebClient (WPF/HTML5) requires the following configuration requirements:

- MS Windows Operating System in Server or Desktop version (with a limit of 10 connections simultaneously)
- The Movicon.NExT I/O Data Server with a sufficient number of WebClient users
- The Windows' "Internet Information Services (IIS)" feature integrated in Microsoft operating systems

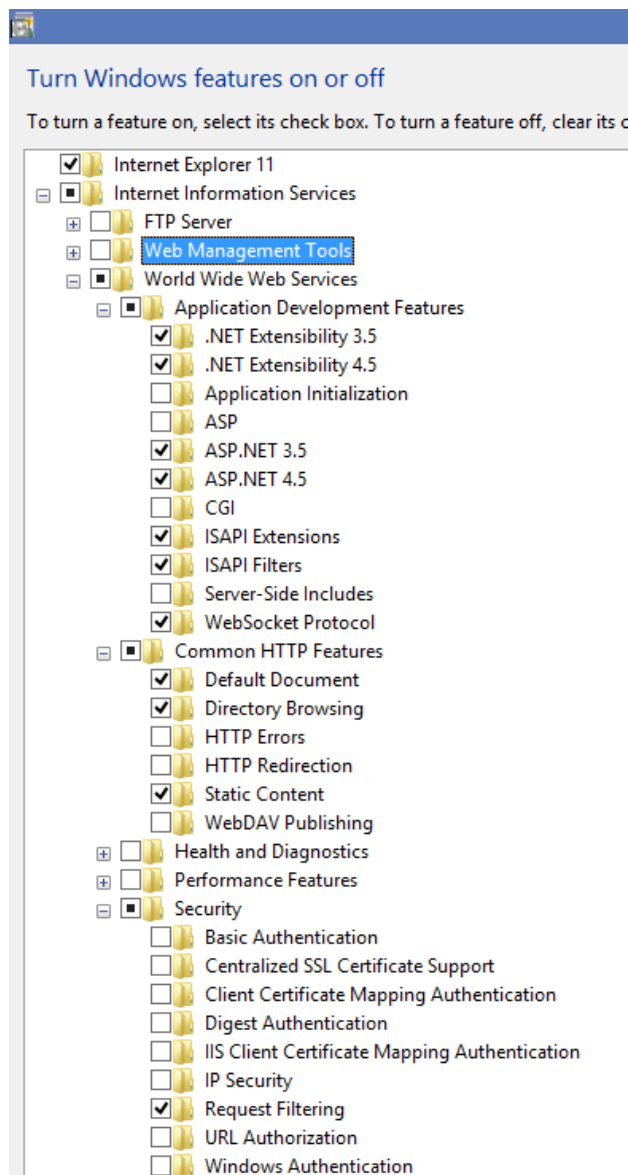
How to configure the "Internet information Services" is shown below and which can be accessed from the Control Panel > Programs and Features > Turn Windows features on or off.

### IIS Configuration for Windows 10 and Windows Server 2016, 2019

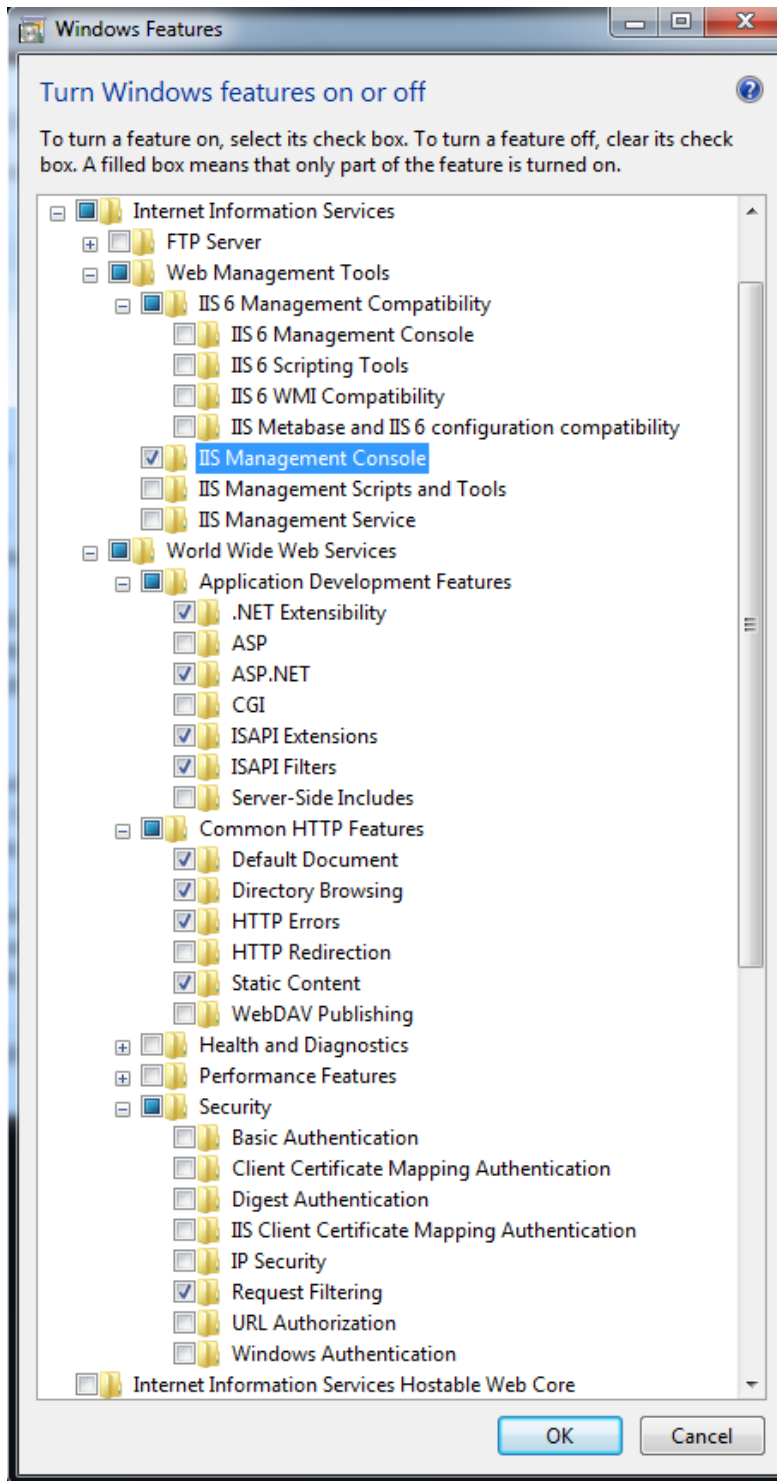




## IIS Configuration for Windows 8, 8.1 and Windows Server 2012, 2012 R2



## IIS Configuration for Windows 7 and Windows Server 2008):



### 14.2.3. Client Limitations

The visualization of graphical pages on the HTML5 Web Client side is limited in how it functions due to its intrinsic Web and Cross Platform features and which must be taking into consideration by the user when designing their project.

Based on the project's needs on the Client side the user should assess whether to use the Web Client or Desktop Client technology or even the Terminal Server technology (Remote Desktop).

Some architectures have both advantages and disadvantages that must be assessed in function with project needs.



**Attention! It is important that the project designer, who is responsible for making the project completely usable by Web Clients, take into full consideration existing web architecture restrictions in order to provide the user with the right functionalities they need.**



To resolve eventual references to the Movicon ToolBox assembly in project scripts correctly, you will need to copy the entire ToolBox folder, contained in the Movicon installation folder (C:\ProgramData\Progea\Movicon.NExT.4.x), to the C:\Windows\System32\inetsrv" folder.

### Web Client side limitations

**Please refer to the Topic on "Technical Specifications" in the Web Client section:**

#### Antivirus

Using an antivirus that has online data traffic analysis functions might slow down the Web Client performances. Therefore it would be best to take into consideration that communication performances may be effected when the antivirus attempts to analyse or filter all the data packets transmitted by online protocols. If there are any problems it is advised to disable the antivirus's online data analysis while using the client (for example, you may have to disable the "online shield" function when using the AVG antivirus).

#### Page Refresh

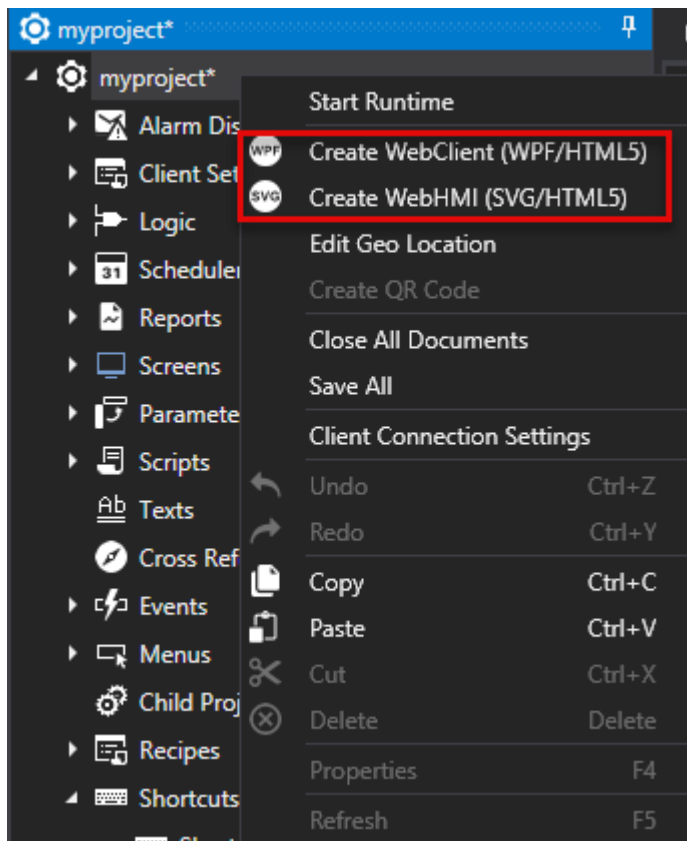
The browser's page "refresh" command (F5), returns to the initial "Tile Page" and restarts the Web Client application.

### 14.2.4. Project Deployment

After having created your project in the Movicon.NExT workspace as well as the Tags and screens as needed, you can choose whether to deploy your application on the Web according to the HTML5 Web Client architecture described previously.

The "Create Web Client (HTML5) command is available from project's Options menu in the Movicon.NExT Workspace or from the toolbar as shown in the screenshot below.





If you are using custom WPF UserControls in screens, in order to use these controls on the WebClient side you will need to copy the relating ".dll" files within this path:

"C:\Program Files\Progea\Movicon.NExT x.x\UFWWebClient.HTML5\bin\Toolbox"

before executing the 'Create Web Client (HTML5) command:



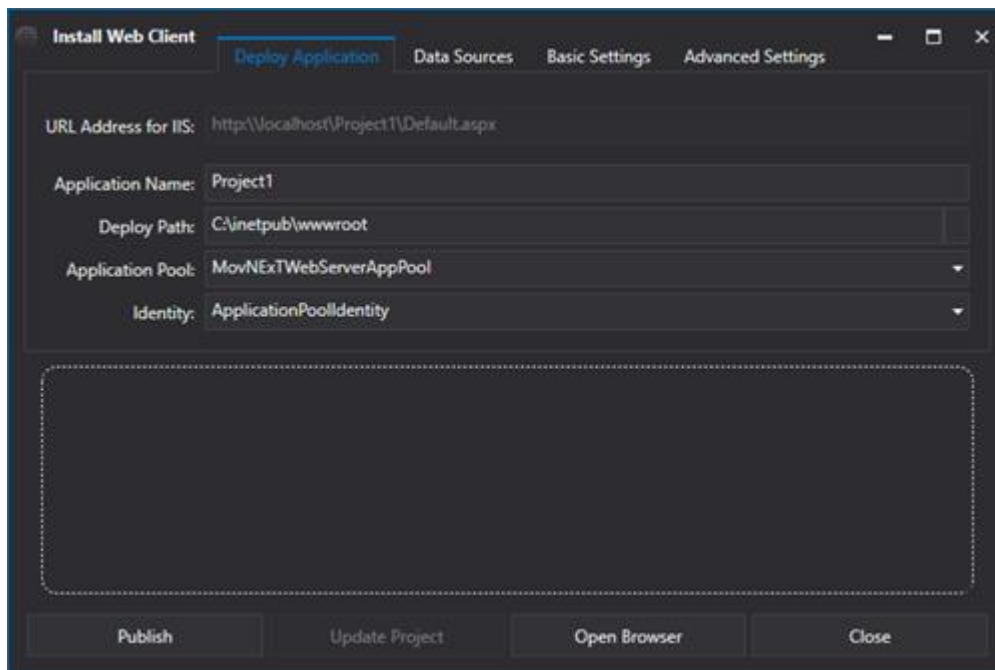
Movicon.NExT can determine whether access to the WebClient is happening from a Mobile device (Smartphone or Tablet) or from a Desktop computer by hijacking the client (Browser or App) on different pages. In order to do this, simply define the screens with the "Mobile" suffix. For example, if your project has two Screen1 and Screen2 pages that can be displayed on Desktop, simply create two new Screen1Mobile and Screen2Mobile screen pages so that navigation from the Mobile Device happens in these pages. It is not necessary to redefine change page commands as the substitution of Desktop pages with those of the Mobile happens transparently.

## Deploy Application window

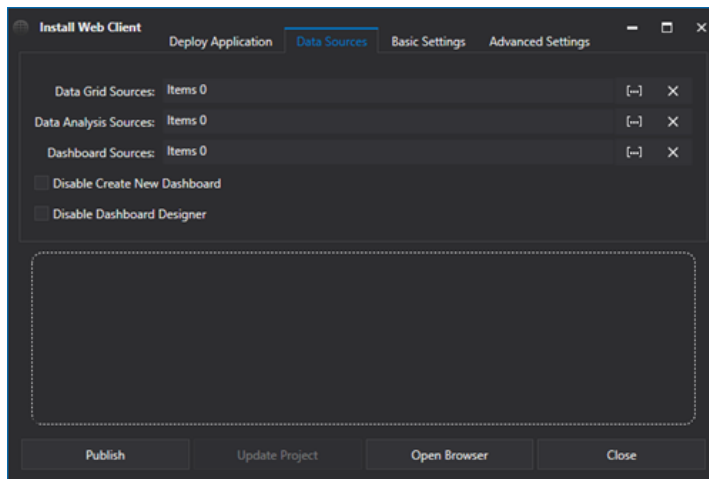
The use of the command to deploy projects on the Web, determines the creation of HTML5 pages and the configuration of the connections relating to the platform's Web Server module.

As a consequence, a data settings window will display, as shown below, to allow you to select, modify or configure the deploy parameters.

The Deploy window permits you to define the following deploy parameters:



<b>Url address for IIS</b>	The Deploy Web address of the server to which the client must point to. Specifies the URL address which must be digitized in the Client browser to display the application's "Main Web Page".
<b>Application Name</b>	Name of the virtual directory to create for the Web Server. It also allows you to replace the name of the deploy URL folder if need be.
<b>Deployment path</b>	The deploy physical path. Indicates the folder in which the deploy web files are available from.
<b>Application Pool</b>	Name of pool with which the web application will be started up. The pool defines the user context within which the Web application is executed and which is necessary, for example, for any eventual access to the SQL Server database managed by the Server project.
<b>Identity</b>	<p>User identity used for the pool. User identity is needed when wishing to access those services outside the project on the server and that request it, such as obtaining access to data on a SQL Server Database. Therefore, if the Web application requests access to a database, it will need to be executed in a pool context with a user identity which is known to the database within which it has been defined.</p> <p>Generally, Identity names created for default are proposed and therefore already exist in the Database as Access Authorization Identity. Otherwise, in order to access the contents of a database over the web, you will need to specify an Identity that has been defined in that database on the Server. To verify and/or enter user identities in a SQL Server database, you will need to run the SQL Server Management Studio.</p> <p>Proposed users are those which are normally standard and default users:</p> <p>"LocalSystem", "LocalService", "ApplicationPoolIdentity", "NetworkService".</p> <p>Otherwise you can indicate a "SpecificUser" (user set as pleased and already defined as one of the users granted access to the DB).</p>



### **Data Grid Sources**

This permits you to set the connection string to the DB which the indicated object will use, in this case the Data Grid, extracting data.

### **Data Analysis Sources**

This permits you to set the connection string to the DB which use the indicated object will use, in this case the Data Analysis, for extracting data.

### **Dashboard Sources**

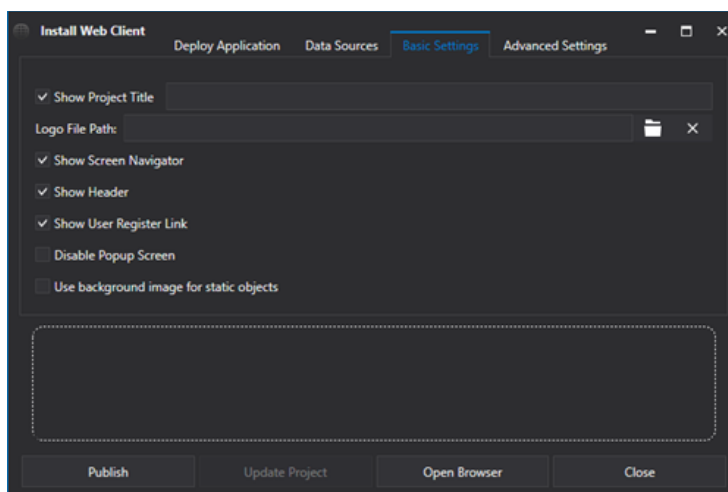
This permits you to set the connection string to the DB which the object will use, in this case the DashBoard, for extracting data.

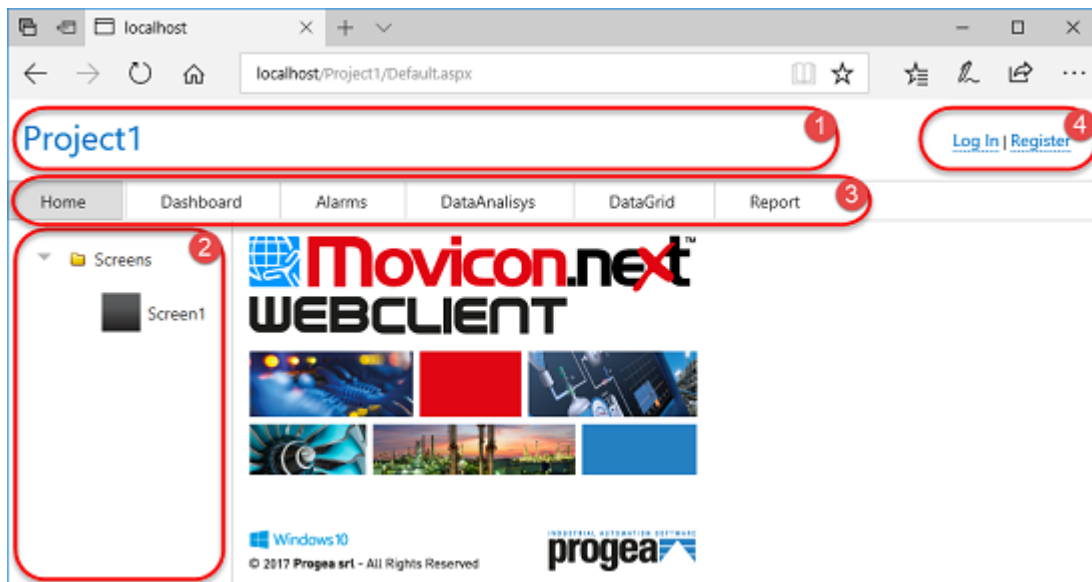
### **Disable Create New Dashboard**

Disables the possibility to create new Dashboards.

### **Disable Dashboard Designer**

Disables the Dashboard editor.





### Show Project Title

Shows the application name in the web page (see point 1 in screenshot above)

### Logo File Path

This property allows you to specify the path of an image to use as a logo in the startup page in Tile Page mode instead of the default Movicon logo. The Logo path must be a path relating to the application. For IIS security reasons no other folder can be accessed unless explicitly configured to do so. Therefore, to simplify things, the image file must be copied manually to the application folder or one of its sub folders. For example, when deploying the 'TestProject', it will be downloaded in the "wwwroot" with the following path:

*C:\inetpub\wwwroot\TestProject*

At this point, if we insert the "CustomLogo.png" image in the application's 'Images' sub folder:

*C:\inetpub\wwwroot\TestProject\Images\CustomLogo.png*

The relative path will be inserted in the Logo File Path property:

*Images\CustomLogo.png*



When deploying the project the folder will be recreated. Therefore, it will be necessary to copy the Logo image file in the path proposed after deploying project.

### Show Screen Navigator

Shows the screen navigator side panel in the web pages (see point 2 in above screenshot)

### Show Header

Shows the toolbar in the web pages (see screen shot in point 3 above).

The HTML page's default menu dashboard can be customized by using the file containing the 'Root.master' menu's definitions that can be found in

the Movicon.NExT "C:\Program Files\Progea\Movicon.NExT 4.1\UFWebClient.HTML5" installation folder.

This allows you to have an ad hoc navigation menu that is already bound to the project's user management (You won't be able to navigate the menu unless you log on) and only allows you to map the commands you wish to use.



Since the root.master file is overwritten in the "C:\inetpub\wwwroot\NomeProgetto" IIS folder, each time you execute a webclient deploy, or copy/paste it manually after each publication, or replace the modified root.master with the one original one in the installation folder, it will automatically be copied from NExT in the publication phase.

**Show user register link**

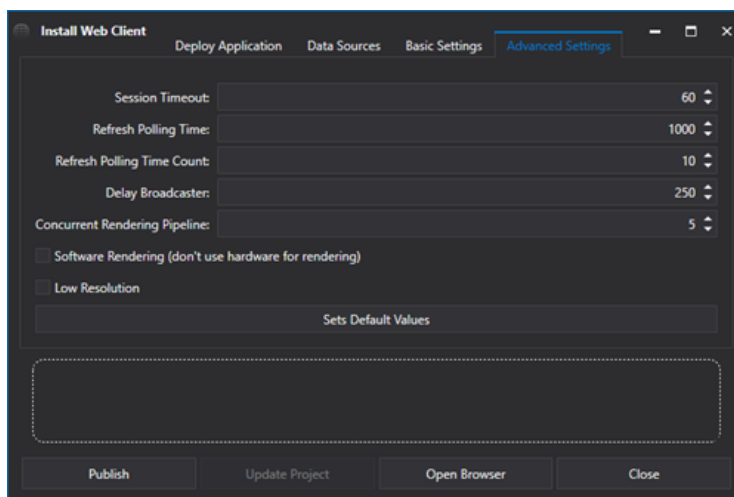
Shows the "Log In|Register" link in the web pages (see point 4 in above screenshot)

**Disable Pop-up Screen**

Disables screens opened as pop-ups.

**Use Background image for static objects**

Activating this option will set all static objects, that are not dynamic, as background image. All the other objects that are dynamic will be handled as usual. In case where you wish to extract a static object from the background image, you will need to activate the 'Force as Dynamic on WebClient' property.



**Session Timeout**

Inactivity time in seconds after which client session will be considered as expired freeing a WebClient license.

**Refresh Polling Time**

Refresh time of dynamic objects on screen in milliseconds. When the "Use background image for static objects" property is activated, only the dynamic objects will be refreshed. Otherwise the whole screen will be refreshed.

**Refresh Polling Time Count**

This represents the number of dynamic objects to be controlled when the time set in the 'Refresh Polling Time' property expires.

<b>Delay Broadcaster</b>	At each 'Refresh Polling Time Count', the web server application will control and notify connected web clients whether each dynamic object has changed position or image or not. This is used to manage broadcasts from the server to the client. Objects that change status are listed and notified to clients after this delay time expires.
<b>Concurrent rendering pipeline</b>	Represents the maximum number of concurrent graphics updates admitted in the web server application for dynamic objects on screens. This value represents a maximum value independently from the number of web clients connects.
<b>Software Rendering</b>	Disables the hardware acceleration to render graphics.
<b>Low Resolution</b>	Uses low resolution images.

The window also has the following command buttons:

<b>Publish</b>	This button is used to publish the HTML5 web applications.
<b>Update project</b>	This button is used to update the published web application when changes are made to the project files.
<b>Open Browser</b>	Test Button that excutes the opening of the published HTML5 web project using the browser predefined in the operating system. The test may require that the data server be started up in runtime.
<b>Close</b>	Exit button.



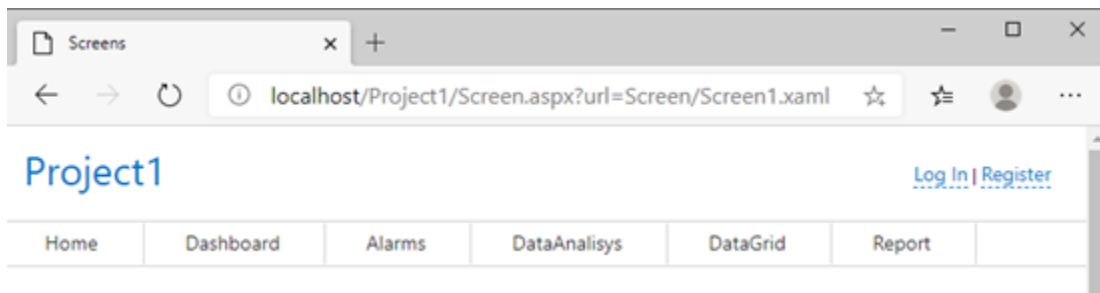
**Attention!** In order to allow the app to connect to the I/O Data Server, at least one of the following conditions must be met:

- The I/O Data Server is installed and started up as Windows Service.
- The I/O Data Server is started up as Administrator (if project is launched from editor, open the editor as administrator).
- Only the net.tcp transport is defined in the I/O Data Server.
- Add the net.tcp transport in addition to the net.pipe in the I/O Data Server and edit the App Name setting with the EndpointRenamed parameter equal to net.tcp://localhost:62846/<app\_name\_di\_progetto>

#### 14.2.5. Web Page menu

The Web Client's Main Page has a menu on the top border which is used for accessing operating commands for some of the server project's features if provided.

The functions are described below:



## Home

This command calls the Web Client's main page. Therefore invokes the return to the main page.

## Report

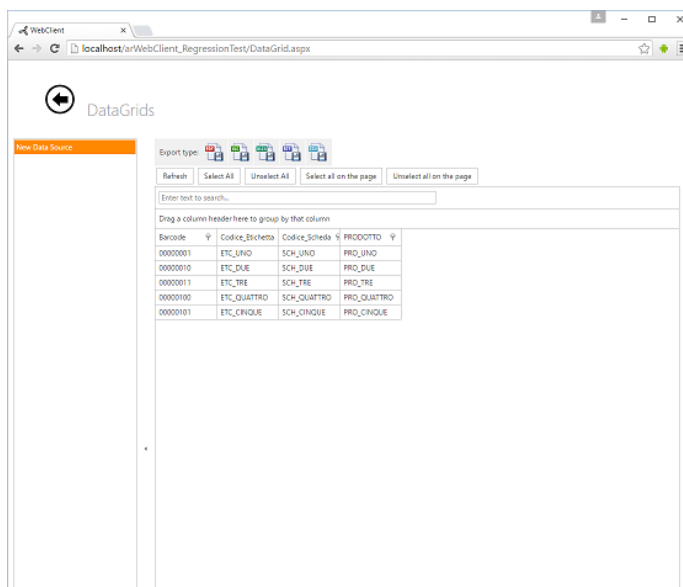
This command is used for viewing Data Report pages on the Client Web side if the Server project has been configured for managing them. This function therefore makes it possible to obtain reports via the web by accessing data recorded on the server to view its contents.

## Alarms

This command calls the project's Alarm page and therefore to view active alarms and to perform operations such as sort by, acknowledge and reset also on the Web Client side.

## Data Grid

The Data Grid object is particularly useful for representing data in your projects in Grid format. When selecting the Data Grid object from the deploy window while using Microsoft SQL as a provider, for example, a dialog window will show allowing you to define the name of the Server and the Database to be used in order to create a connection string.

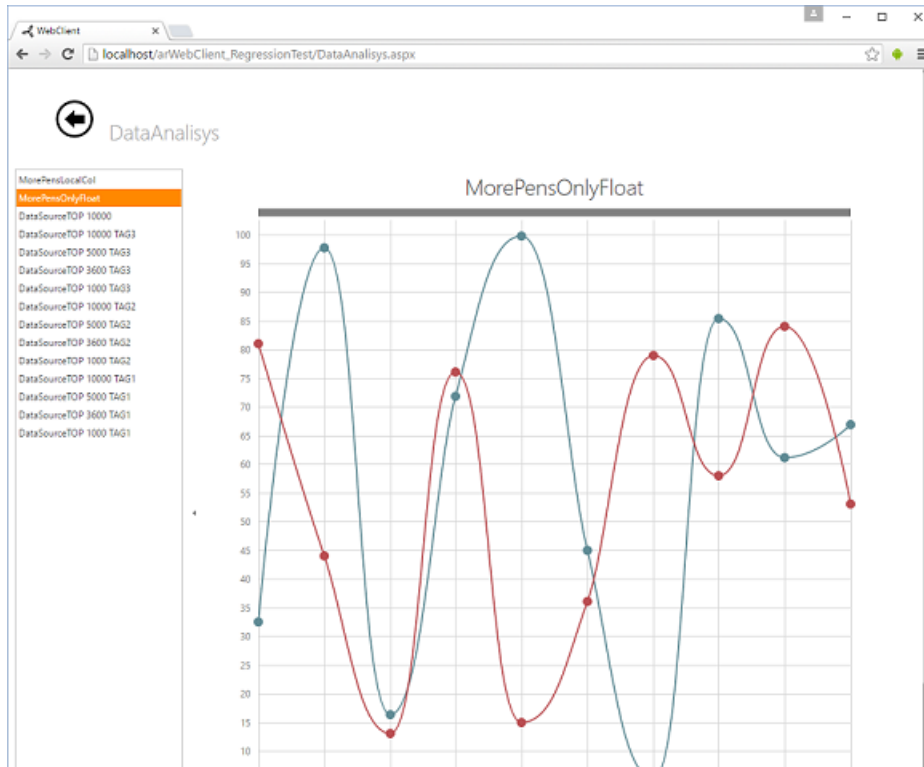


## Data Analysis

The Data Analysis objects are managed in the same way as described above but with 2 exceptions that need to be considered during the deployment procedures:

- the first parameter in the select field (where the query is to be inserted) must always be a time column.

- The fields after the query will be used to represent the chart's curves.

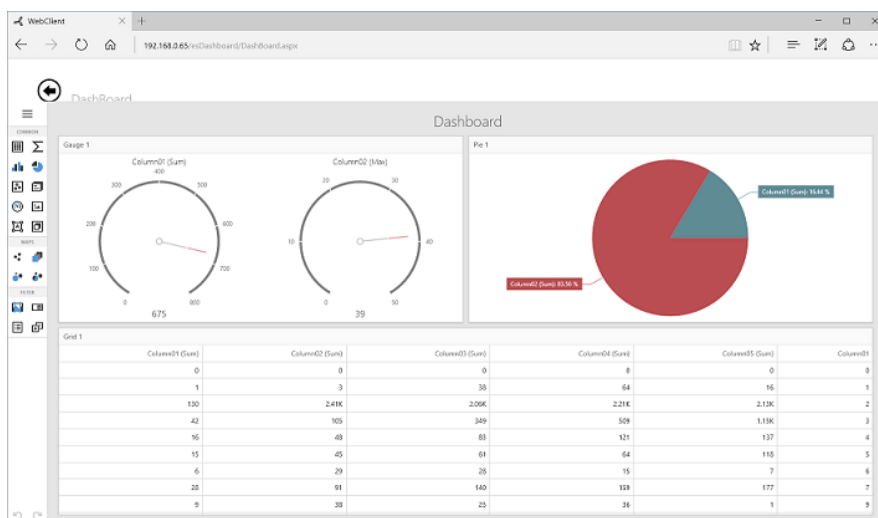


There are two calendar controls underneath the 'Date range picker' caption at the top of the page which can be used to apply a filter on data relating to the DataAnalysis within the time range defined in the initial query.

Data is loaded when the fetch button is pressed.

## Dashboard

The DashBoard object is used to create a page or pages where different objects can be inserted (eg. Chart, Grid and Gauges) and connected to data sources. These Dashboards can also saved and reused for further use.



The option to create/edit new Dashboards has been included with the program's 3.1 version. By means of making a simple modification to the web.config file, as indicated in the examples below, new Dashboards can be created or the option allowing this can be disabled.



```

<appSettings file="">
  <clear />
  <add key="Uri" value="C:\inetpub\wwwroot\arTestDashboard\Project\arTestDashboard.UFProject" />
  <add key="ClientSessionName" value="arTestDashboard" />
  <add key="RefreshPollingTime" value="100" />
  <add key="RefreshPollingTimeCount" value="10" />
  <add key="DelayBroadcaster" value="250" />
  <add key="SessionTimeout" value="60" />
  <add key="ConcurrentRenderingPipeline" value="5" />
  <add key="DisableStaticOptimization" value="False" />
  <add key="DisableCreateNewDashboard" value="False" />
  <add key="Theme" value="Blend" />
  <add key="ValidationSettings:UnobtrusiveValidationMode" value="None" />
  <add key="ChartImageHandler" value="storage=file;timeout=20;dir=c:\TempImageFiles\" />
  <add key="StaticOptimization" value="False" />
</appSettings>
<connectionStrings>
  <remove name="LocalSqlServer" />
  <add name="LocalSqlServer" connectionString="Server=(local);Database=Movicon.Membership;Integrated Security=SSPI" />
</connectionStrings>

```

`<add key="DisableCreateNewDashboard" value="False" />` in this case it will be possible to create new dashboards

`<add key="DisableCreateNewDashboard" value="True" />` disables the possibility to create new dashboards

#### 14.2.6. Managing Client Users

The Movicon.NExT supports a security management for Client users using a user authentication logon procedure. This means that if the Project Server has been set with this security management any users wishing to access the project from the Web Client will have to logon according to User rights and password defined in the Server project.

In order to do this the **"Enable User Manager"** property from the project's User Management must be enabled.

In this case the project will be deployed configuring the web.config as indicated:

```

<authorization>
  <allow users="?" />
</authorization>

```

which denies access to website to all unauthorized users.

While if "False"

```

<authorization>
  <allow users="*" />
</authorization>

```

allows access to all users.



**Warning:** the user level management currently behaves differently on the desktop client: if a user has a lower level than the one set on the HTLM5 Client, the command objects that require a higher authentication lever will not be made visible.

## User Management for database access

In addition to the Users management for accessing project command defined on the server, it may also be necessary to configure and define access to databases from the Web Client side.

As web application performs direct access to the Database without receiving data from the server, the web application must be "acknowledged" and authorized by the SQL Server database.

It is for this reason that when a web application is deployed, the Pool and Identity are defined to determine the web application's execution and the user identities must be among those which possess user rights and access right to the SQL Server instance.



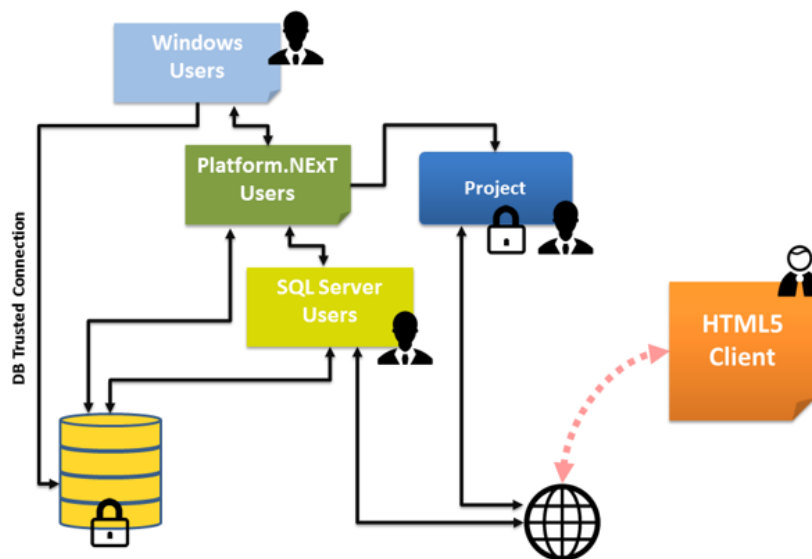
To display the contents of a SQL Server database, which is managed by the Movicon.NExT Server project, on the Web Client side, the web application will need to be run in a Pool with identities known and present in the SQL Server DB.

When the project is deployed with the default settings being:

- Application Pool = MovNExTWebServerAppPool
- Identity = ApplicationPoolIdentity

You will need to give access rights to "IIS APPPOOL\MovNExTWebServerAppPool" users in the database.

See the parameters that can be set when deploying the project from the Server on the web.



*This shows the main concepts of the user management in applications.*

After having enabled the user management, you can use the web to register as a new user by using the register link and access the project pages to display them.

The new user is created with the predefined settings with user level = '0' and with the bit mask completely deselected.

If you wish to inhibit the creation of new users by registering them over the web, you can modify the web.config as follows:

```
<system.web>  
<authorization>
```

```
<allow users="User name to be authorized"/> //several users can be inserted to allow them access by
separating their names with a comma
<deny users="*" /> // this denies access to those users that have not been explicitly indicated on the list
beforehand.
</authorization>
```



Once the web.config file is protected and inaccessible by web, it will not be possible to modify it with the web client.

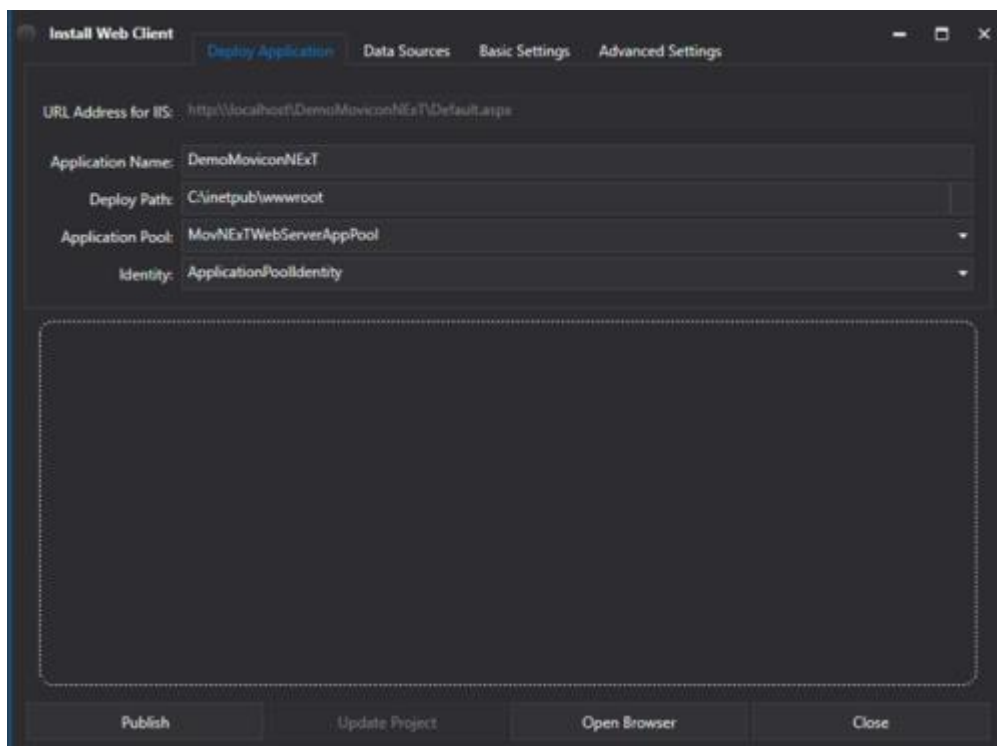
### SQL Database Access using the WPF/HTML5 WebClient feature

To display the contents of a SQL Server database, managed by the Platform.NExT Server project on the Web Client side, the web application must be executed in a Pool with user's identity and present in the SQL Server database.

When the project is distributed with predefined settings:

- Application Pool = MovNExTWebServerAppPool
- Identity = ApplicationPoolIdentity

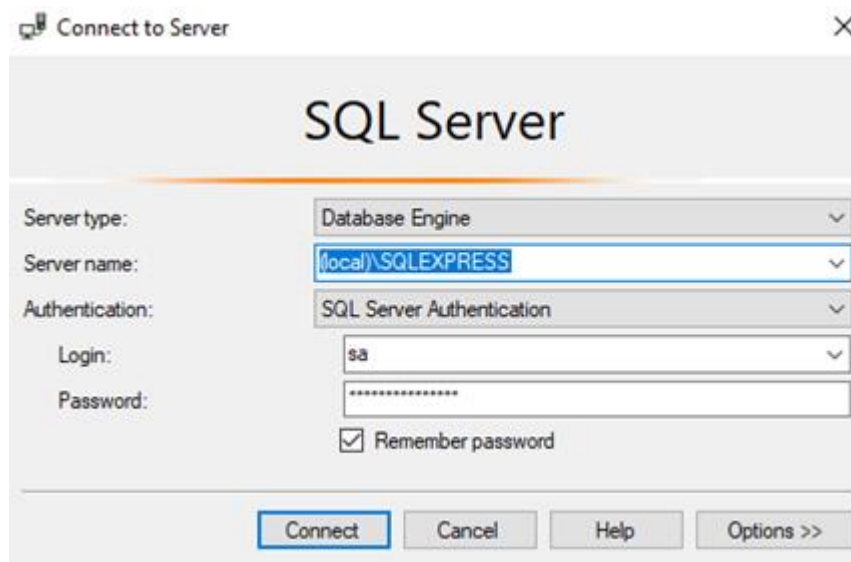
As we can see in the Deploy window:



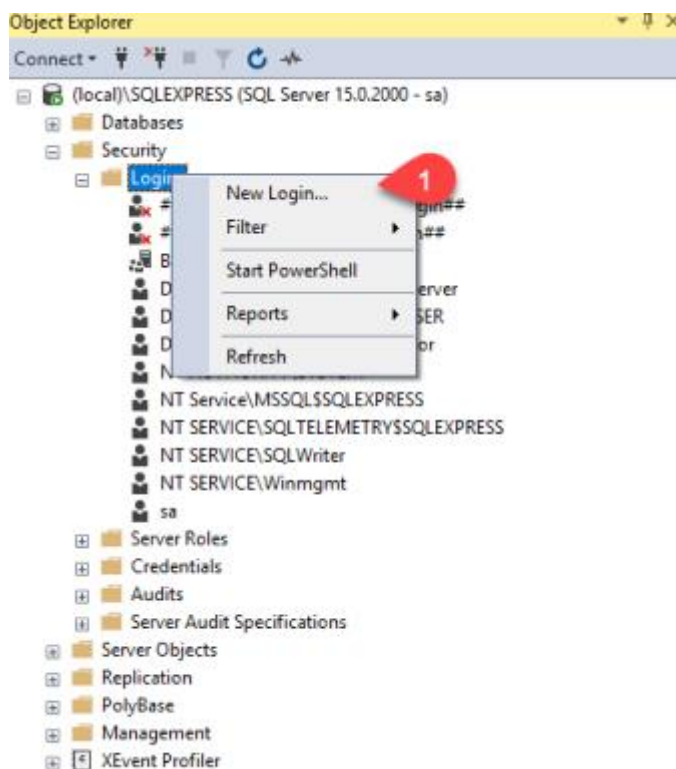
It will be necessary to grant access rights to the "IIS APPPOOL\MovNExTWebServerAppPool" user in the database.

The simplest way to do this is to use an external tool such as Microsoft Management Studio. To activate the necessary authorizations, proceed as follows:

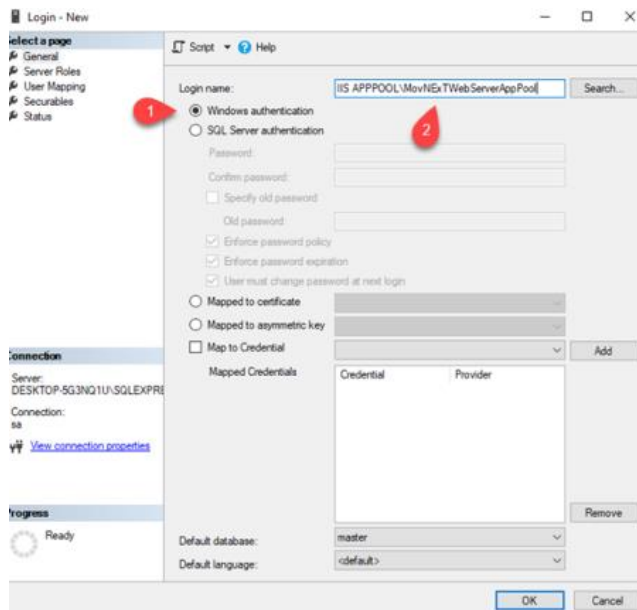
1. Open Microsoft Management Studio and access with SysAdmin credentials



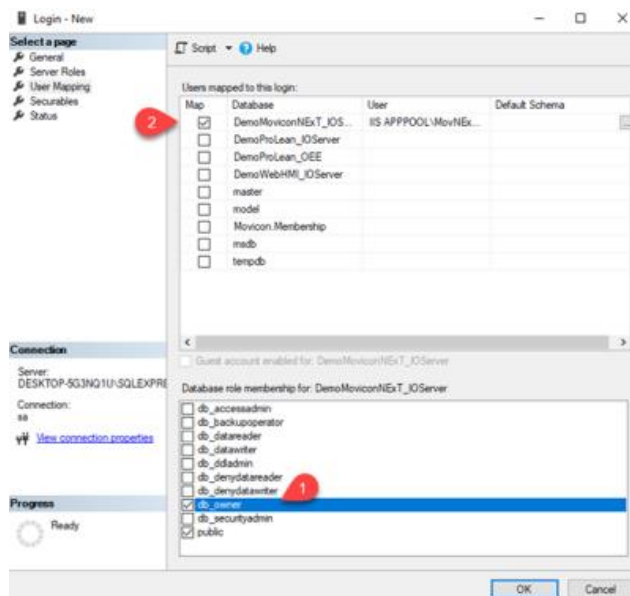
2. Open the Security\Logins folder in the Object Explorer on the left, then right click and select New Login



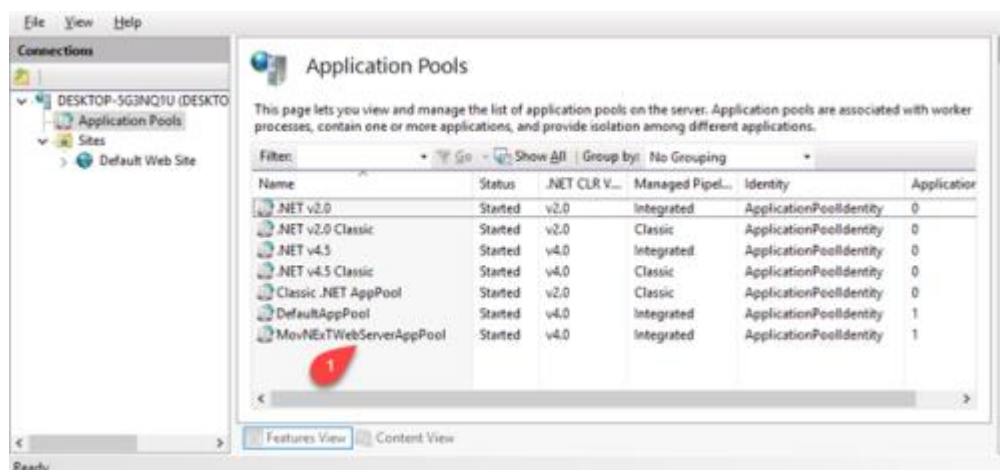
3. Create the new SQL Server user: IIS APPPOOL\MovNExTWebServerAppPool



and assign db\_owner to the desired database:



4. The MovNExTWebServerAppPool must exist in the Internet Information Service (IIS)



It is created when a Movicon NExT web client application is started up for the first time after distributed.

#### 14.2.7. Client Errors

In event of disconnection towards a "<ObjectName> - OPC UA Session Keep Alive bad status>" error message will be reported in the objects displayed on the Web Client.

This error indicates that the Server connection is not available. Therefore you will need to check whether the Server address is reachable and that the project on the Server has been started up in runtime correctly.

## 14.3. WebClient (APP)

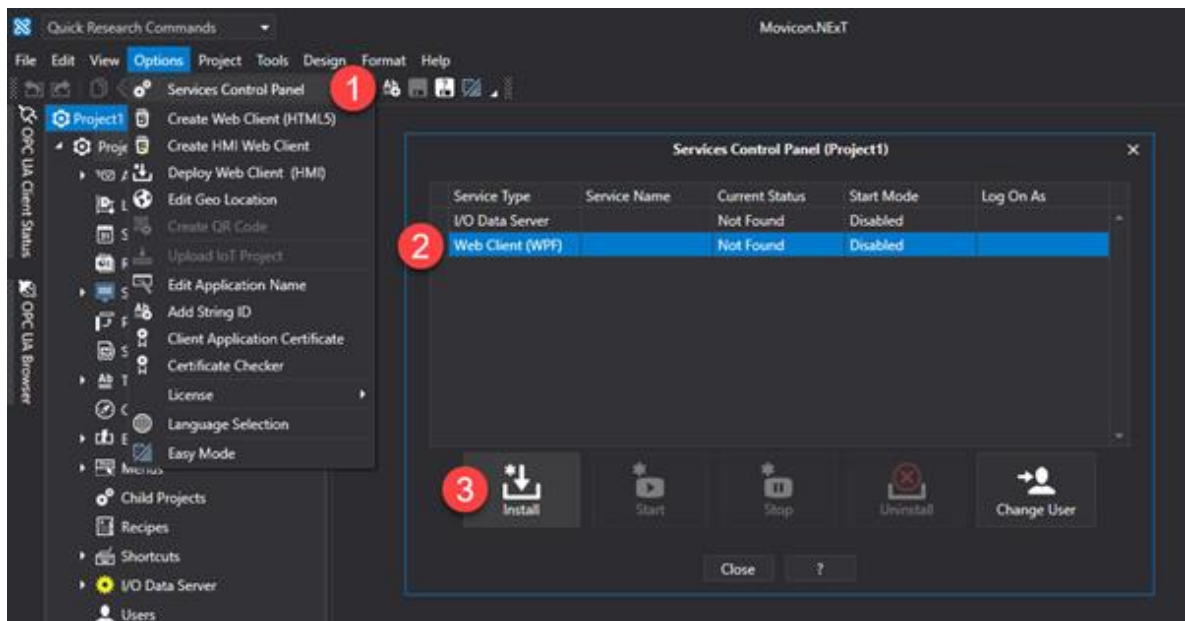
### 14.3.1. Deploying Projects

Movicon.NExT offers an app that you can use as a remote client to connect to your application's screens in a completely independent way rather than using Desktop client or HTML5.



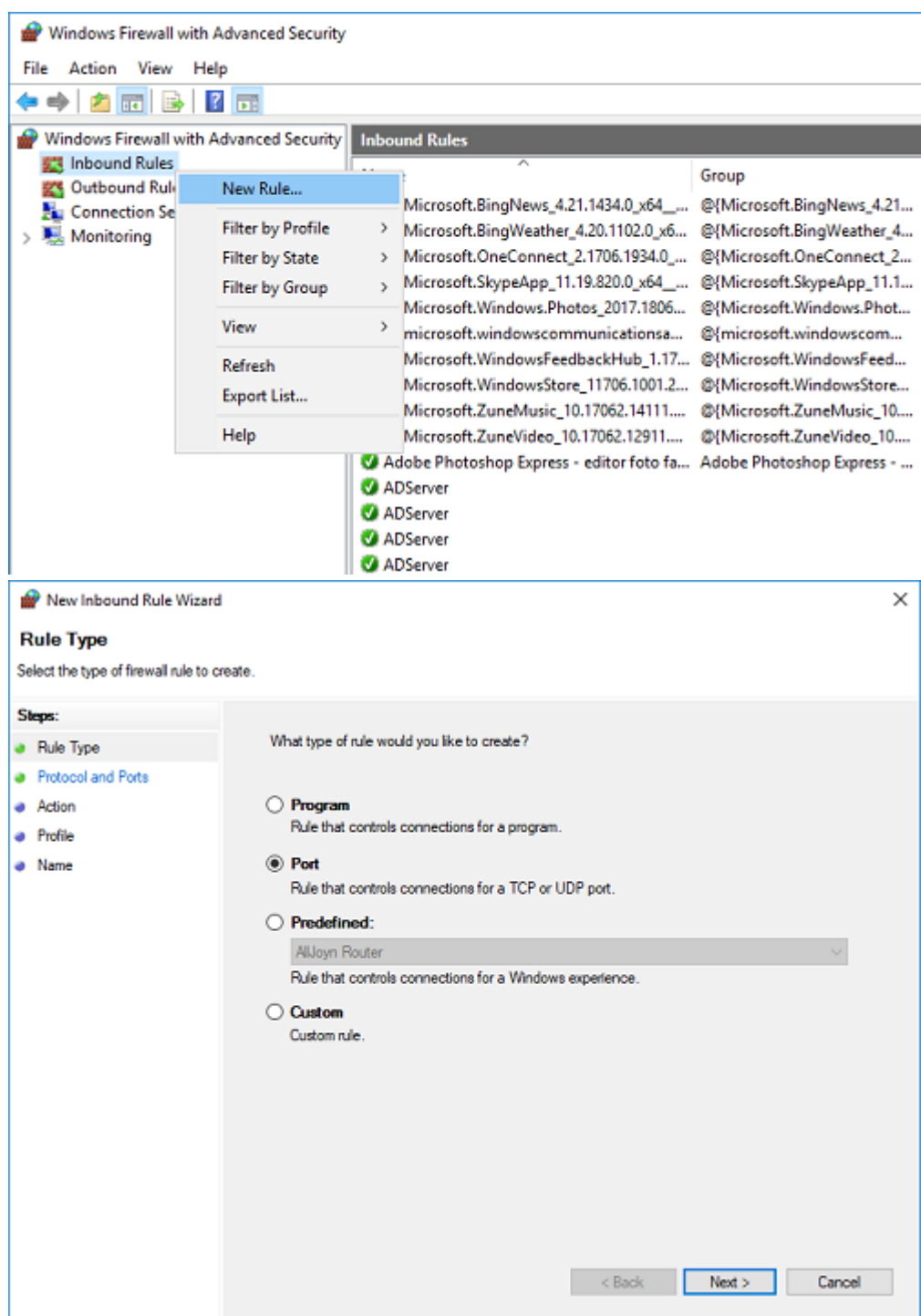
The Movicon.NExT Apps use third party open source components.

In order to permit the app to connect to the NExT Server that deployed the project, you will need to install and launch the "Web Client (WPF)" service from the "Service Control Panel" as shown in the image below:



Once the service has been installed and launched, it might be necessary to configure the active Firewall or Antivirus to allow the use of the 8089 port (for default) for the TCP protocol.

The Windows Firewall should be configured as follows:



New Inbound Rule Wizard

Protocol and Ports

Specify the protocols and ports to which this rule applies.

Steps:

Rule Type

Protocol and Ports

Action

Profile

Name

Does this rule apply to TCP or UDP?

☒ TCP

☐ UDP

Does this rule apply to all local ports or specific local ports?

☐ All local ports

☒ Specific local ports:

8089

Example: 80, 443, 5000-5010

< Back

Next >

Cancel

New Inbound Rule Wizard

Action

Specify the action to be taken when a connection matches the conditions specified in the rule.

Steps:

Rule Type

Protocol and Ports

Action

Profile

Name

What action should be taken when a connection matches the specified conditions?

☒ Allow the connection

This includes connections that are protected with IPsec as well as those are not.

☐ Allow the connection if it is secure

This includes only connections that have been authenticated by using IPsec. Connections will be secured using the settings in IPsec properties and rules in the Connection Security Rule node.

Customize...

☐ Block the connection

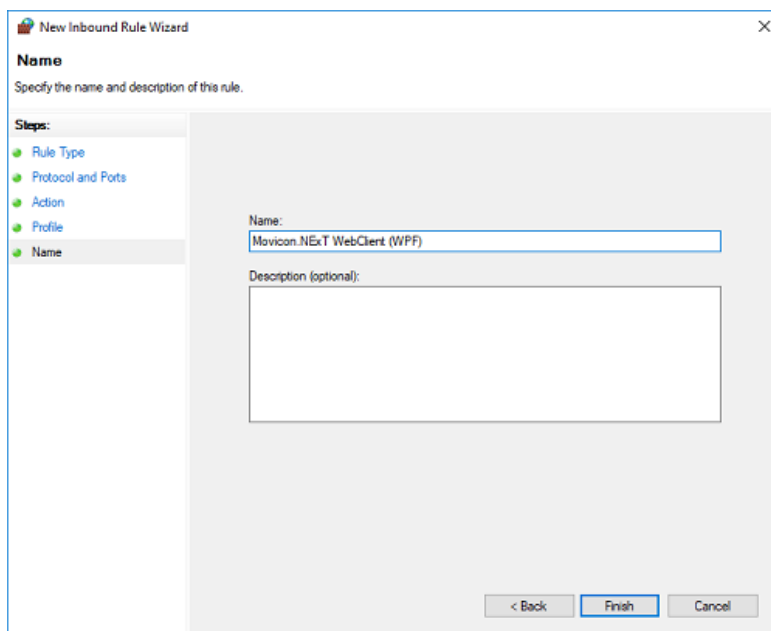
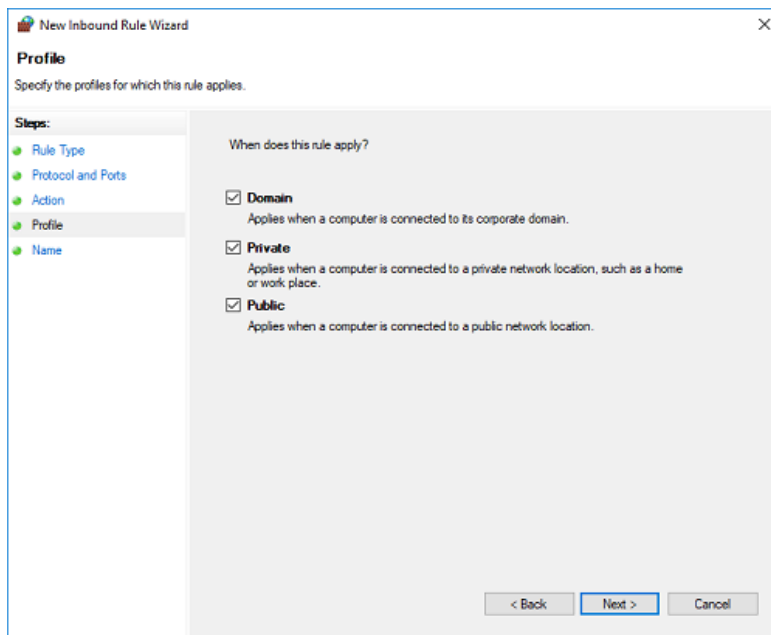
< Back

Next >

Cancel

1024





Attention! To allow the app to connect to the I/O Data Server, the following conditions must be met:

- The I/O Data Server must be installed and started up as Windows service.
- The I/O Data Server must be started up as Administrator (when launching project from editor, open editor as administrator).
- Only set the net.tcp as transport type in the I/O Data Server.
- Add the net.tcp transport type in addition to net.pipe in the I/O Data Server and edit the App Name Settings with the EndpointRenamed parameter equal to net.tcp://localhost:62846/<app\_name\_of\_project>



Attention! When the project involves the use of Users, you will need to enable the user used for starting up the Web Client (WPF) service between the sysadmin of the SQL Server instance.

The user predefined for the Web Client (WPF) service is "LocalSystem" identified in SQL as "NT AUTHORITY\SYSTEM"

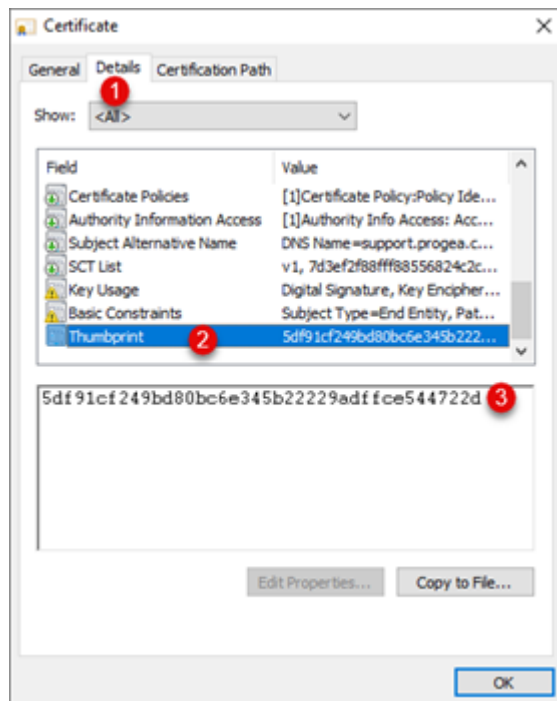
## Using HTTPS certificates for the App service

The App service can be added by means of using HTTP on the default port 8089 or HTTPS on the default port 8090.

If you use HTTPS, you will need to:

1. obtain the HTTP certificate with a private key, generated for the hostname that the app service will expose (e.g. demonext.progea.com)
2. add the certificate to the Trusted certificates of the PC on which the service of the Apps is running
3. Using the command prompt, associate the certificate to the port on which the service will listen (the default port is 8090), with the following command: `netsh http add sslcert iport=0.0.0.0:8090 appid={12345678-db90-4b66-8b01-88f7af2e36bf} certhash=Thumbprint_certificate.`

Where "Thumbprint\_certificate" is replaced with the string which can be retrieved from the certificate's properties, on the Details tab (as shown below).

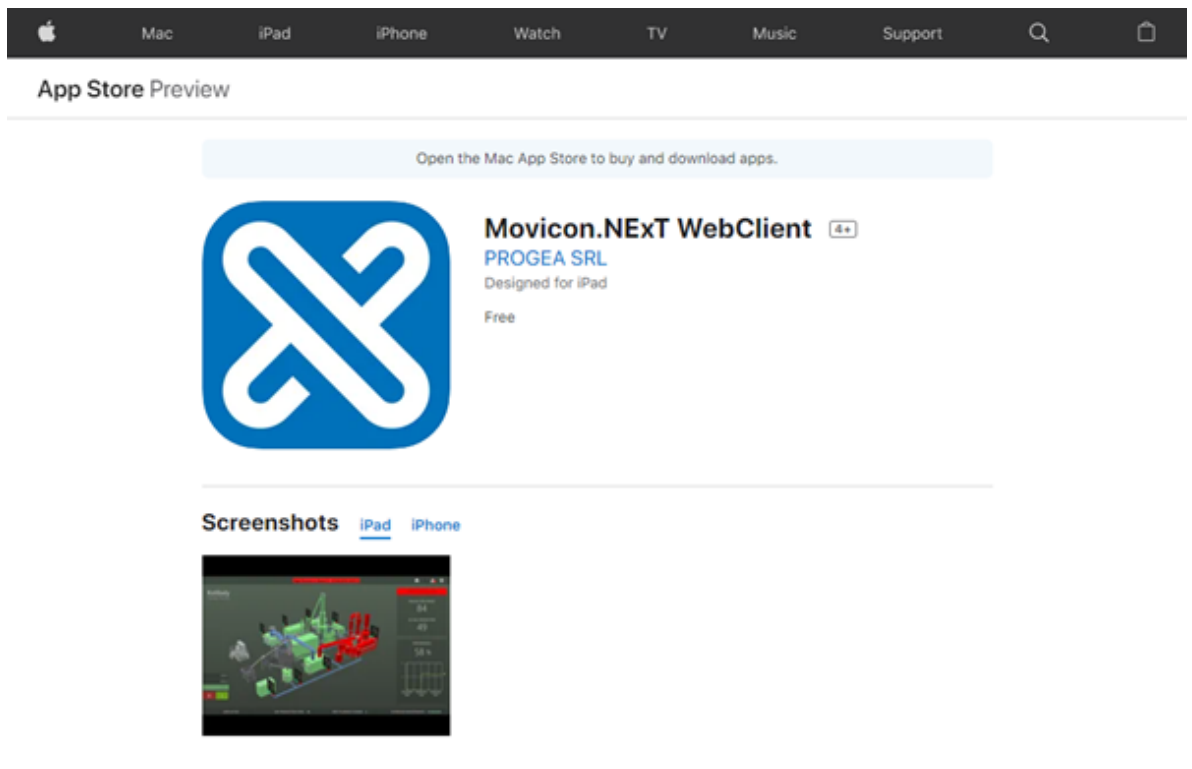


## 14.3.2. APP Installation and Configuration

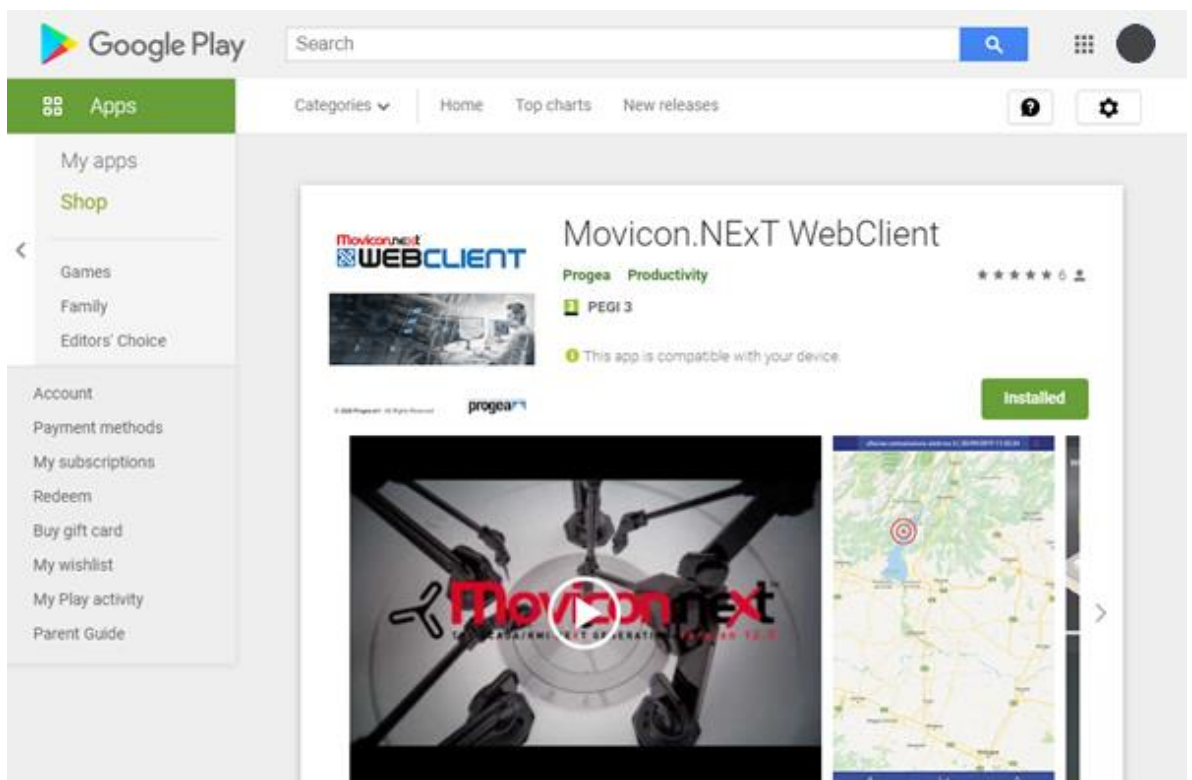
### APP installation

The Movicon.NEXT WebClient app can be installed on tablets, smartphones and wearable devices (e.g. the Hololens) by accessing the below stores:

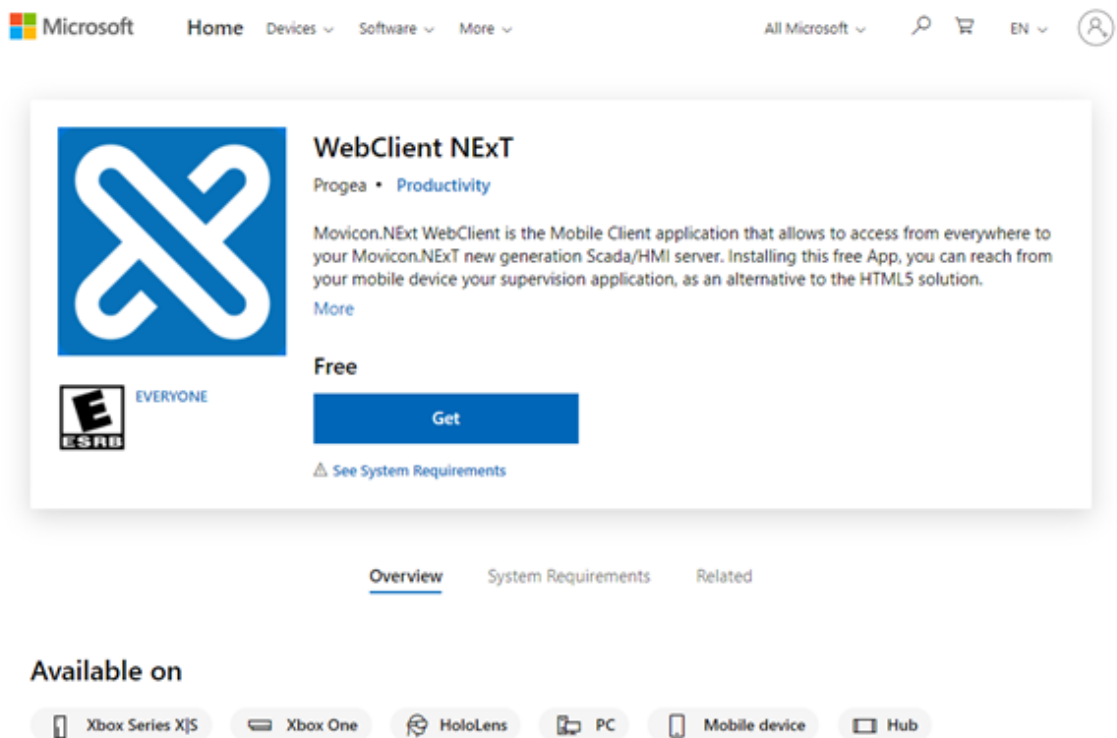
App Store for iOS devices



Play Store for Android devices

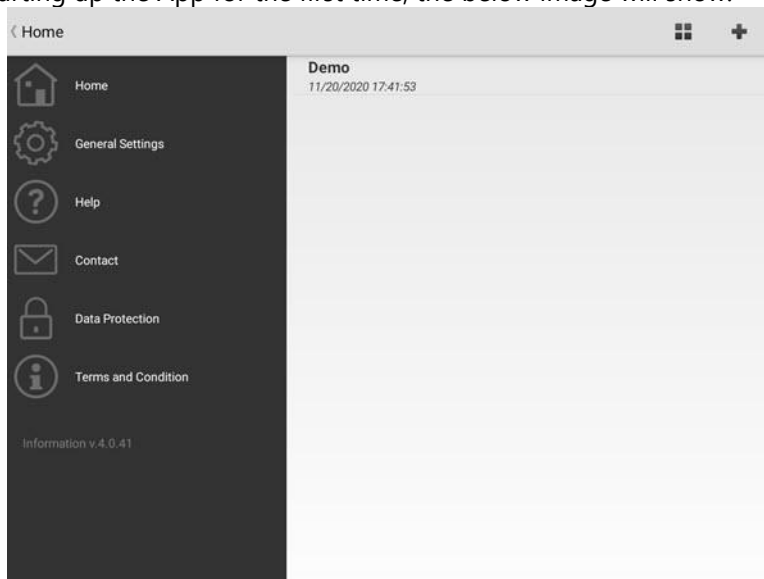


Microsoft Store for Microsoft devices



## App Configuration

When starting up the App for the first time, the below image will show:



The following menus are presented on the left:

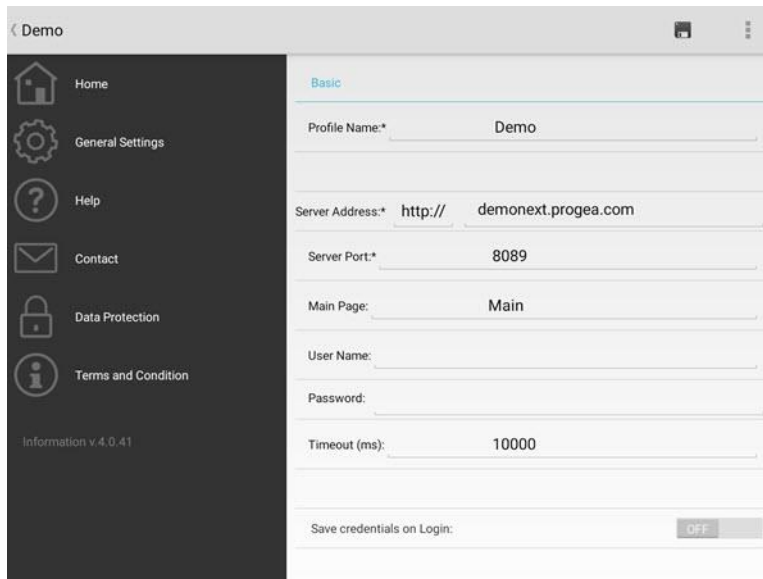
- **Home:** this allows you to access and configure the connection profiles.
- **General Settings:** this allows you to configure the app's general settings.
- **Help:** a brief guide that shows you how to manage the profiles.
- **Contact:** this allows you to send an email to Technical Support.
- **Data Protection:** this shows the cookies and privacy policies.
- **Terms and Condition:** shows the app's terms and conditions of use.

At the top on the right you will find:

- A button for sorting profiles (e.g by creation date),
- A button to add a new profile as shown in the below screenshot.

To edit or delete an existing profile, simply long press the profile.

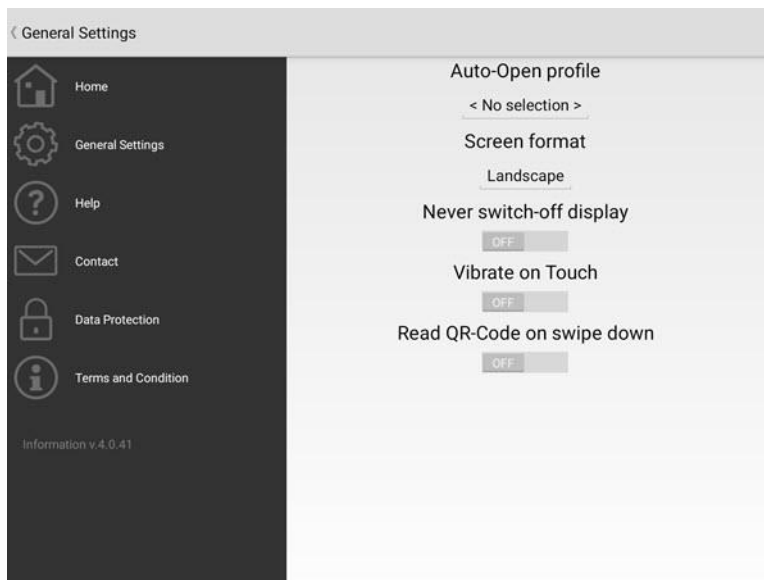
The will show a default profile with the name of "Demo" that will allow you to test the use of the app with a project published online by default.



Le impostazioni di un Profilo di connessione sono:

- **Profile Name** = the name to assign to the profile
- **Server Address** = protocol (http or https) to use and the server address to connect to. When you use the HTTPS protocol, you will need to obtain a certificate issued by a Certificate Authority (CA) and follow the steps indicated in the "Using HTTPs certificates for the App Service" paragraph in the "Deploying Projects".
- **Server Port** = indicates the TCP port on which the Apps' service listens on. The default port for HTTP connections is 8089, while the default port for HTTPS is 8090.
- **Main Page** = indicates the project page which will open at startup.
- **User Name/Password** = username and password to use for accessing the project when user management is active in project. When username and password are not specified, a login window will show at startup.
- **Timeout** = indicates the connection timeout to the Apps' service in milliseconds.
- **Save Credentials on Login** = allows the username and password to be saved when used in the Login window.

Within the "General Settings" window shown below:



you will be able to define other parameters which include:

- **Auto Open Profile** = drop-down menu with which the user can select the profile that will open automatically when starting up the app.
- **Screen Format** = screen display format
- **Never Switch Off Display** = this allows the Display to stay always active in the apps where there is no user interaction happening. This can only be used in IOS systems (however, the device settings will override this property type).
- **Vibrate on Touch** = activates vibration when touched (however, the device settings will override this property type).
- **Read QR-Code on Swipe down** = this is used to activate the device's camera for reading QR code by using the "swipe down" command.

### 14.3.3. App Limitations

#### App Limitations

Visualizing pages with the app has some functional limitations that the design engineer should take into consideration when designing their project. They must evaluate whether to use the App or HTML technology based on their specific project needs.

Each architecture has its own advantages and disadvantages which must be taken into account accordingly:

App Advantages:

- Service integrated in NExT no needing Microsoft IIS
- The use of specific apps for mobile devices

HTML5 Advantages:

- Use of Microsoft IIS with option to use redundant infrastructure for multiple Web Client service.
- Any Browser can be used.



Attention! If the design engineer wants to make the project fully usable by app, they must take special care that such limits do not prohibit users from using the necessary functions correctly.

#### WPF Web Client side Limitations (App)

Please refer to the the Web Client WPF (App) section in the topic on "Technical specifications".

## 14.4. WebHMI (SVG/HTML5)

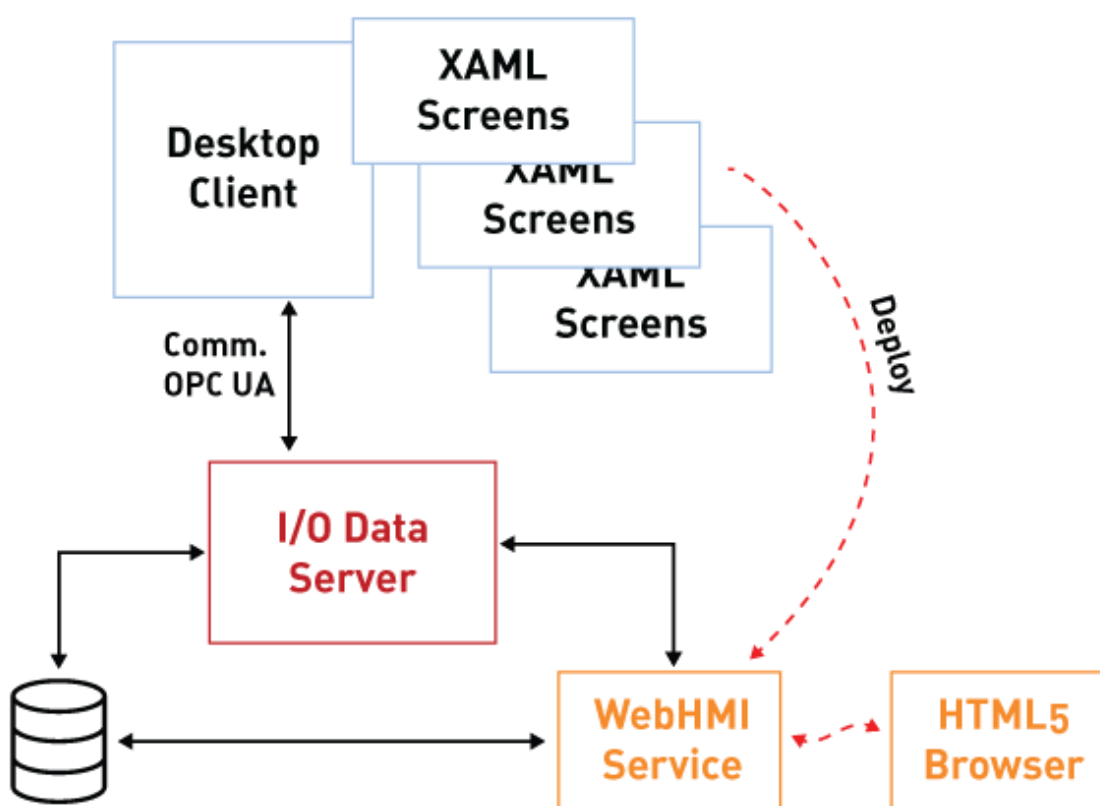
### 14.4.1. Introduction to using the WebHMI

WebHMI is a solution to use browsers to visualize HMI projects created with Movicon.NExT.

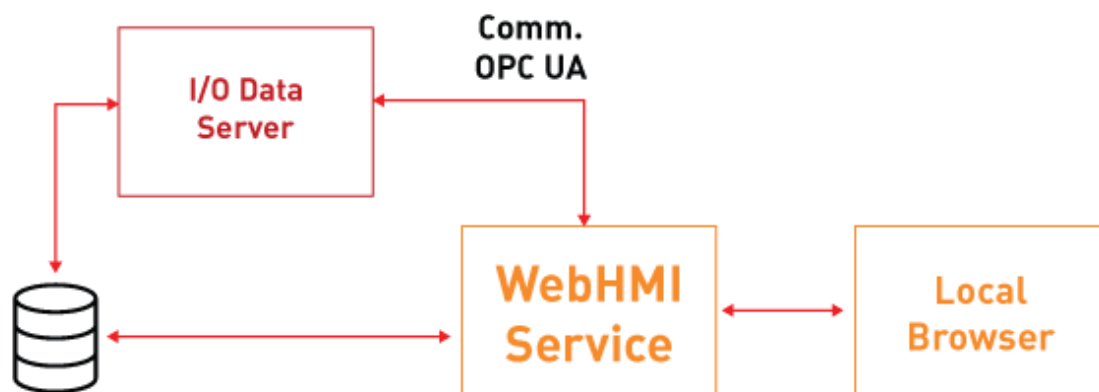
The new WebHMI Web Client makes project pages available in SVG/HTML5 format using an integrated Web Server which interfaced with the Movicon.NExT I/O Data Server through OPC UA.

The two main fields of application can be distinguished as follows:

1. WebHMI Client :To expose projects created with Movicon.NExT for access via HTML5 Browsers as an alternative to WPF/HTML5 WebClient.



2. WebHMI Runtime: As stand-alone applications on the HMI (Windows or Linux) that use the I/O Data Server and displays graphical pages locally using a browser opened at full screen.



In respect to the previous WPF/HTML5 Web Client solution, the WebHMI which could also be called SVG/HTML5 has distinct characteristics as described below.

### Using SVG graphics

The vector graphics used by Symbols and User Controls in Movicon.NExT are WPF/XAML type. Therefore, the project screens created with XAML vector graphics are converted to SVG in HTML5 web pages when the project is deployed directly from the Movicon.NExT editor.

### Processing on Client side (Browser)

Originally the IIS Web Server was responsible for processing graphical pages displayed by connected Client Browsers within the WPF/HTML5 Web Client architecture. This meant that the size of IIS Web Server had to be quite substantial in order to support more than one Client Browser connection at the same time.

This has all changed with the new SVG/HTML5 Web Client architecture. Now, the integrated Web Server only makes HTML5 pages available to Browsers and uses the I/O Data Server to manage communications. Therefore, graphics are now processed on the Client (Browser) side giving the integrated Web Server leeway to support a higher number of Client connections simultaneously.

### Support to GeoScada and Augmented Reality functions

These two features are now available with the new WebHMI to allow use of Geographical Maps on par with those of the Client Desktop along with the use of Augmented Reality in combination with the Speech Recognition used in objects.

Thanks to these new additions, the WebHMI solution is now ideal to use as an alternative to the Client Desktop or Web Client (WPF/HTML5) and as a stand-alone solution on Windows and Linux operating system based HMI user panels.

### Supported Browsers

Movicon WebHMI has been test run on different platforms:

- **Windows:** Chrome, Edge.
- **iOS:** Safari, Chrome;
- **Android:** Chrome.

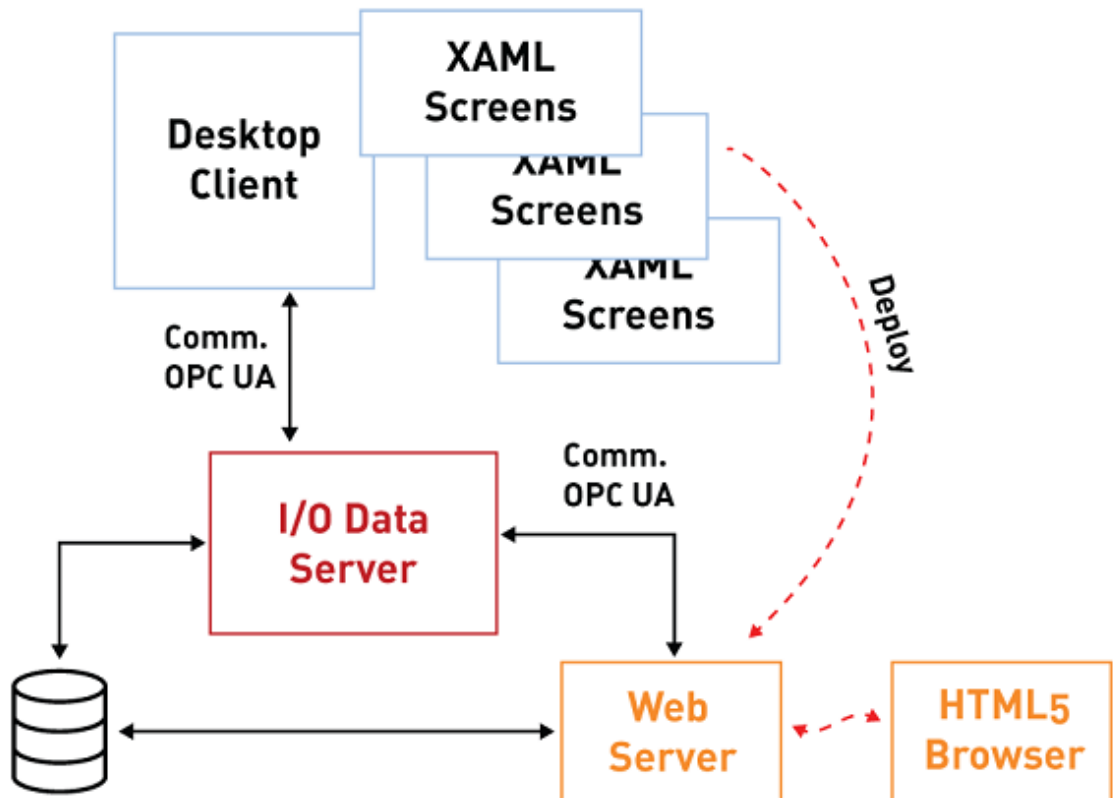
Support and the proper functioning of Movicon WebHMI cannot be guaranteed for all other browsers, such as Mozilla Firefox for example.



#### 14.4.2. WebHMI Client

As seen in the introduction, the WebHMI uses an integrated Web Server to expose projects for access by Client Browsers.

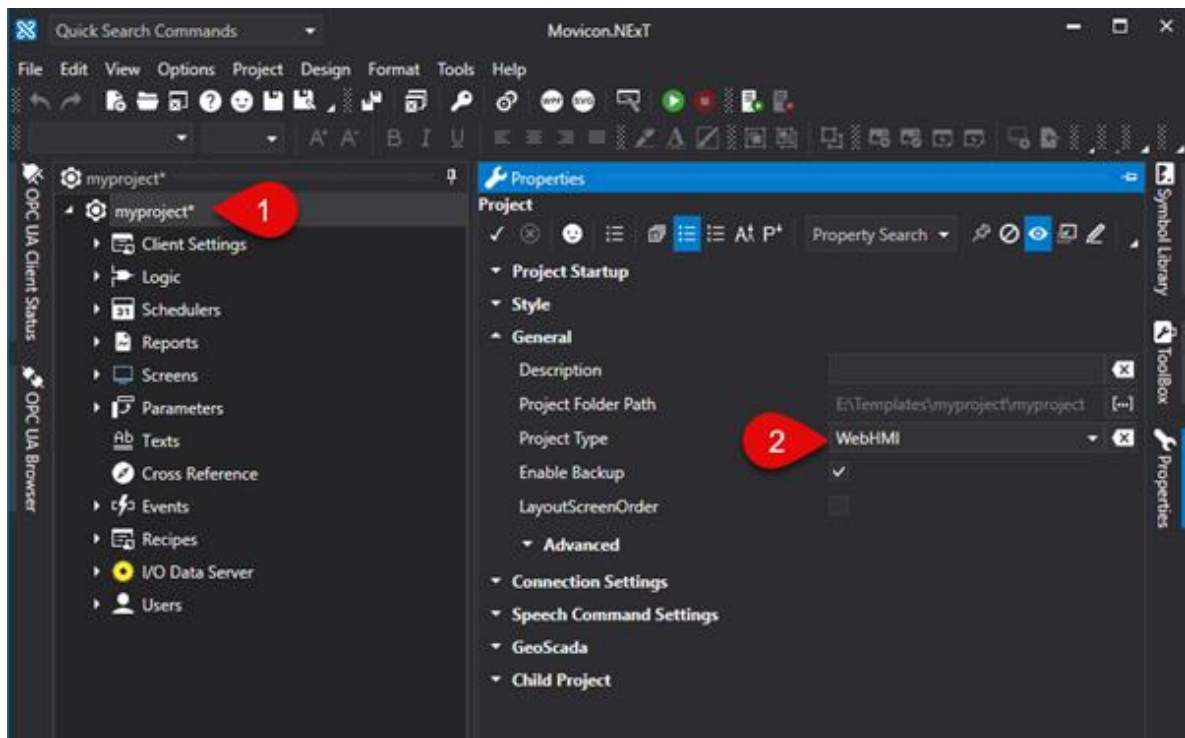
In order to deploy the Movicon.NExT project using the integrated Web Server, its pages are exported from XAML vector graphic format to SVG format using the deploy tool as shown in the below paragraphs.



#### Propriety: Project Type

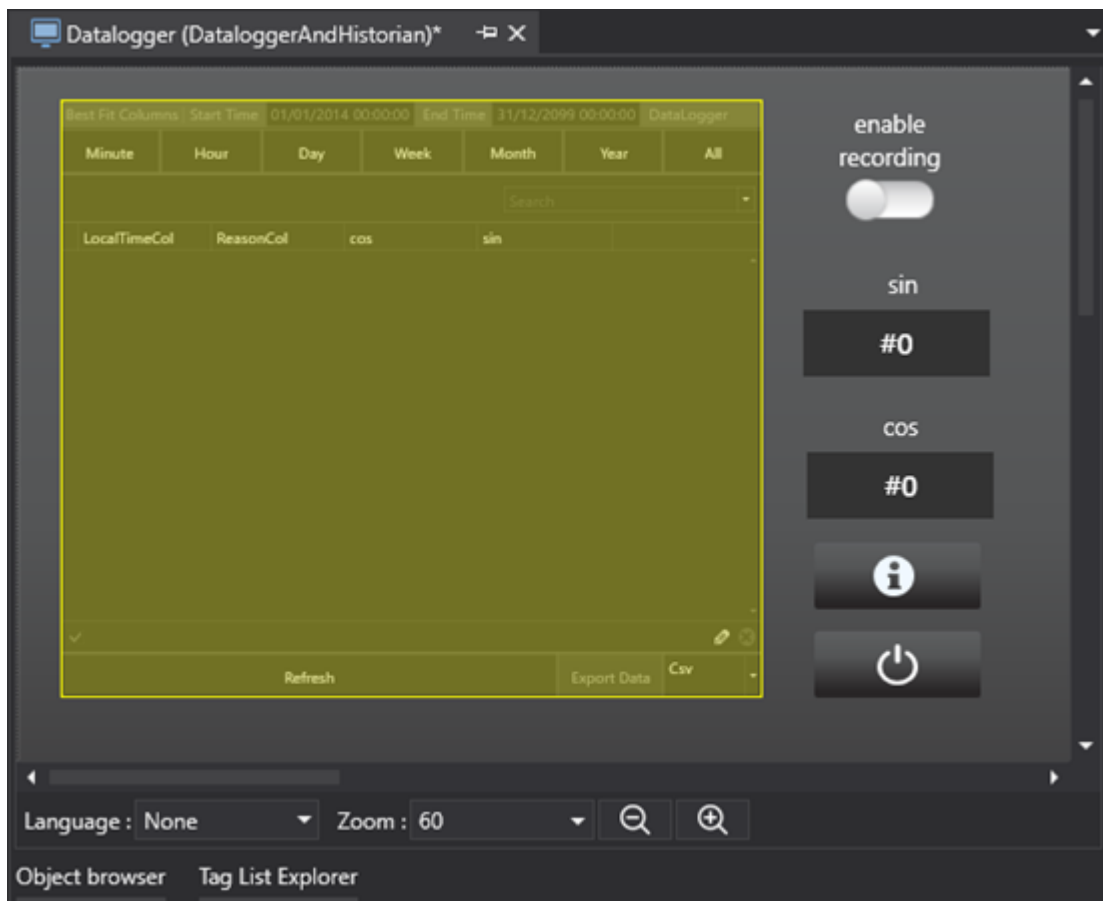
The "Project Type" property is one of the project properties that is used to set the project as a "WebHMI" type.

When setting the project as "WebHMI", the Project Explorer, Toolbox and Animation Explorer and Command windows will only show the resources, animation objects and command that are supported while visualizing and using Web Client WebHMI.



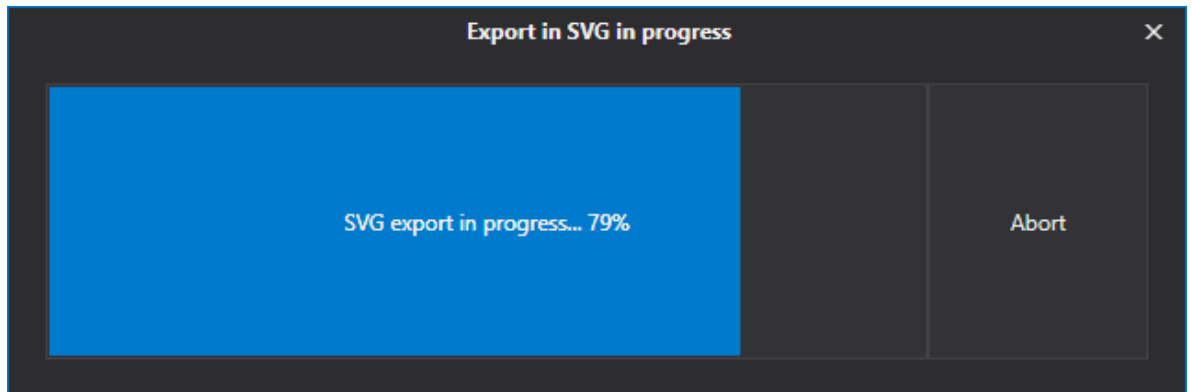
This property setting is particularly useful when:

- planning to develop a project that includes only those functions supported by WebHMI,
- needing to verify the compatibility of a previously developed project with WebHMI visualization. Any objects and symbols used in the project and not supported by WebHMI will be highlighted as shown in the screenshot below.

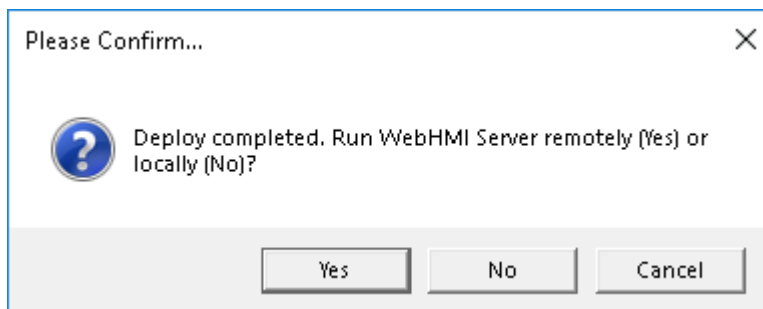


### Deploying Projects Locally

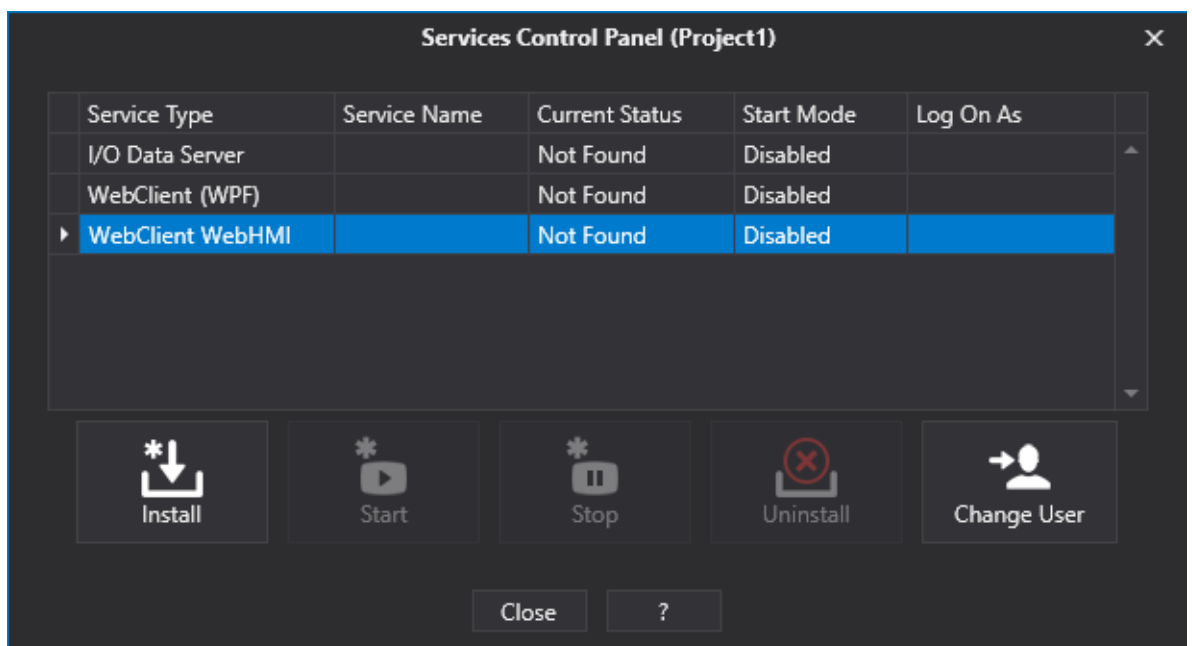
To deploy a project in the integrated Web Server to then make it accessible to Client Browsers, simply select Options menu > Create WebClient (SVG/HTML5) or select Create WebClient (SVG/HTML5) from the project's Toolbar to start the project's export to SVG.



Once this procedure has terminated, you will be prompted to confirm whether or not the Web Server and the Project is to be deployed and run on a remote HMI as shown below:



When selecting "No", a Service Control Panel will open through which the "WebClient (SVG/HTML5)" service can be installed and run locally:



### Connecting with Browser

The project is deployed by the integrated Web Server on ports 5000 (http) and 5001 (https).

The default configuration requires that the requested received on port 5000 be automatically redirected to port 5001. To change these settings please refer to chapter on "Integrated Web Server Settings" .

The following addresses can be used to access pages exposed by the Web Server:

```
http://<WebServer_IP_Address>:5000
https://<WebServer_IP_Address>:5001
```

Example: `https://192.168.10.10:5001`

It is also possible to open a different screen to use as the start page instead of the one defined as the Main Page in the project. To do this simply add the name of the screen to be opened to the url using the following syntax:

```
https://<WebServer_IP_Address>:5001/?page=<ScreenName>
```

where <ScreenName> is the name of the Screen without an extension. If the screen is inside a folder, the name of that folder should also be specified.

Example:

```
https://192.168.10.10:5001/?page=Screen1
https://192.168.10.10:5001/?page=Folder1/Screen1
```

Ultimately, a parameter file can also be passed while loading the specified screen. In order to do this, you must add the name of the parameter file to the url after the name of the screen:

```
https://<WebServer_IP_Address>:5001/?page=<ScreenName>&parameter=<ParameterName>
```

where <ParameterName> is the name of the Parameter file without an extension. If the parameter file is inside a folder, you should also specify the name of that folder.

Example:

```
https://192.168.10.10:5001/?page=Screen1&Parameter1
https://192.168.10.10:5001/?page=Folder1/Screen1&Folder1/Parameter1
```

In addition, the WebHMI page can be opened by performing user logon directly from the URL. To do this, simply add the username and password to the URL using the following syntax:

```
https://<WebServer_IP_Address>:5001/?user=<UserName>&psw=<UserPassword>
```

dove <UserName> is the name of the user who will logon and the <UserPassword> is the user's password.

Example:

```
https://192.168.10.10:5001/?user=User1&psw=12345
```

User logon can also be performed using a specific screen as follows:

```
https://192.168.10.10:5001/?page=Screen1&parameter=Parmeter1&user=User1&psw=12345
```

### 14.4.3. Integrated Web Server Settings

The WebHMI's integrated Web Server settings can be changed by editing the "appsettings.json" file which is found in "C:\Program Files\Progea\Movicon.NExT 4.0\WebHMIServer\"

The customizations available are listed below in the following paragraphs.

### Changing the listening port (default 5000 e 5001)

The default listening port can be changed by modifying the values of the "HttpListeningPort" and "HttpsListeningPort" parameters.

### Deactivation of the http or https protocol

When necessary, it is possible to select and set the respective parameter to zero of either the http or https protocol to deactivate it.

For example: "HttpsListeningPort": 0

### Deactivating the redirection from port 5000 to port 5001 (active for default)

The redirection from port 5000 to port 5001 can be deactivated by adding the "DisableHttpsRedirection" parameter to the "WebNExTHMISettings" group and setting it to True come as shown in the image below.



### Using a customized SSL certificate

A customized certificate can be used in substitution of the automatically generated self-signed certificate by adding the following rows to the "appsettings.json" file:

```
"Kestrel": {
  "Endpoints": {
    "HTTP": { "Url": "http://*:5000" },
    "HTTPS": {
      "Url": "https://*:5001",
      "Certificate": {
        "Path": "certificate_file.pfx",
        "Password": "certificate_password"
      }
    }
  }
}
```

where "certificate\_file.pfx" is the name of the certificate including the absolute path.

It is possible to use a relative path with respect to the "WebNExTHMI.exe" executable, as in the case of the example.

#### 14.4.4. WebHMI Functions and Limitations

Below are listed the features supported by WebHMI. Any features not shown on this list are not supported.

##### Supported Objects from the Toolbox

- Alarms: Alarm Banner, Alarm Window (without use of filters), Historian Log Viewer
- Progress Bars
- Buttons: standard and with styles
- DB Connectors: Combo-box, Grid Control
- Dialog Boxes: Check-box, Combo-Box, Spin button
- Displays
- Arc Bargraphs: all
- Gauge: circular e transparent 180°
- Meter
- Gauge Advanced
- Hot Regions
- Dynamic Images
- Lights: all
- Bargraphs: all
- Bargraph Advanced: all
- Draws: all
- Recipes: Recipe Viewer
- Schedulers: Scheduler Viewer
- Selectors: 2 position selectors, 3 position selectors (0-1-2 e 1-0-2)
- Sliders: horizontal, vertical, flat horizontal, flat vertical
- Switches: Push button switches, toggle switches
- Screen Tabs
- Texts: Animated Text, Text Box
- Trends: Chart, Trend Data Analysis, Real Time Trend, State Bar Chart, Sparkline Chart
- Viewers: Report Viewer, PDF Viewer
- Historian Data Viewers: Data Logger Viewer, Historian Viewer



The "viewer" objects of the Toolbox, such as "Alarm Window", "Event Log", "Grid", "Data Analysis Trend", etc., on the WebHMI side, can be presented with a reduced set of available commands compared to the Client Desktop version. In principle, the following may be missing:

- some commands of the object toolbar
- some command buttons
- filter functionality
- data export or print functionality
- pop-up windows (e.g. Palette window or Data Analysis Grid)

##### Supported Project Resources

- Logic
- Schedulers
- Reports
- Screens
- Parameters
- Texts
- Cross Reference
- Events (excluding Events linked to System tags or local Client tags. They are supported only Events linked to I/O Data Server Tags)
- Recipes

- IO Data Server
- Users (excluding Domain users, and it is not possible to share / synchronize users between clients of different types)

### Supported Animations



Animations are only supported by the symbols from the Symbol Gallery and objects from the "Draws" group in the Toolbox.

- Background Color
- Visibility
- Blink
- Filling
- Height
- Width
- Scale
- Rotation
- Move X
- Move Y

### Supported Commands

- Screens (except for the Auto Close and Monitor ID options)
- Tag Value
- Open Pad
- Recipes (except for commands: Export/Import and Show Recipe)
- Users
- Change Language
- Reports
- Alarms
- Text to Speech
- Open Map

### Other limitations

- The .NET Core framework support is needed in stand-alone applications (WebHMI Runtime)
- Only USB hardware or software licences are supported in stand-alone applications (WebHMI Runtime), and not NET licenses.
- vb.net scripts are not supported on both server and client sides in stand-alone applications (WebHMI Runtime), but logic is available.
- Only scripts on server side run by the Script service are supported when using WebClient only (SVG/HTML5).
- Projects cannot be saved on database.
- The I/O Data Server redundancy is not supported.,
- Data validation is not available for data recorded with the "Audit Trail" (CFR21),
- 3D graphics are not supported.
- Multitouch functions are not supported.
- "opc.tcp" and "http" are the only client-server communication transports available.
- Log files are saved as text files and not xml. files.
- "tiff" images are not supported.



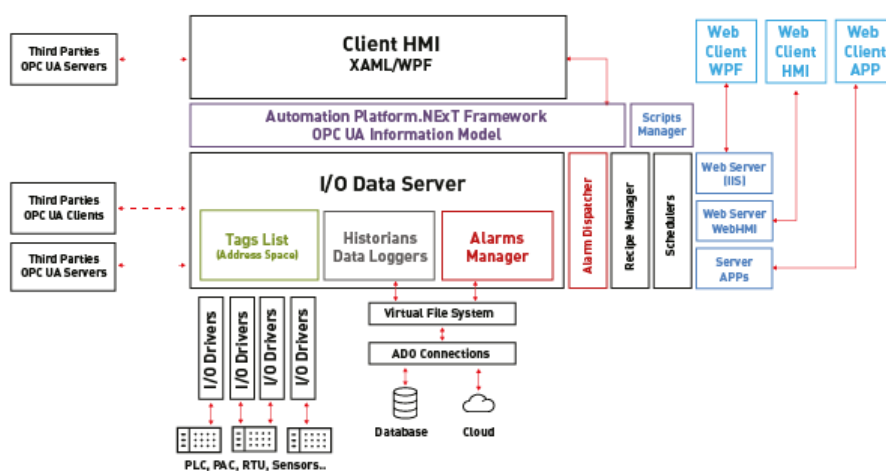


# 15. OPC UA

## 15.1. OPC UA Information Model in Platform.NExT

The Movicon.NExT framework bases its data information model on the OPC UA technology, both for differentiating the Client-Server infrastructure and in the complete development its functional parts:

- The I/O Data Server performs the function of Data Access (DA),
- The Alarm Server fulfills the role of Alarms and Conditions (AC)
- The Historian Server manages Historical Access (HA)



This exclusive technology, which is totally transparent to users using the framework, offers the great advantage of using an open, expandable and modular platform that conforms to standards and consents full integration of simple or complex data models deriving from other levels of automation such as IEC61131-3 PLCopen control systems or modern MES/ERP systems.

### OPC Unified Architecture (UA) IEC 62541

The OPC Unified Architecture (UA) was released in 2008 as an independent platform based on service-oriented architecture that integrates all the functionality of the individual OPC Classic specifications in one unique and extensible framework. This multi-layered approach makes it possible to achieve the original design specification goals of:

- Functional equivalence: all COM OPC Classic specifications are mapped to UA.
- Platform independence: from an embedded micro-controller to cloud-based infrastructure.
- Secure: encryption, authentication and auditing.
- Extensible: ability to add new features without affecting existing applications.
- Comprehensive information modeling: for defining complex information.

#### Functional Equivalence

building on the success of OPC Classic, OPC UA was designed to enhance and surpass the capabilities of the OPC Classic specifications. OPC UA is functionally equivalent to OPC Classic, yet capable of much more:

- Discovery: find the availability of OPC Servers on local PCs and/or networks.

- Address Space: all data is represented hierarchically (e.g. files and folders) allowing for simple and complex structures to be discovered and utilized by OPC Clients.
- On-demand: read and write data/information based on access-permissions.
- Subscriptions: monitor data / information and report-by-exception when values change based on client's criteria.
- Events: notify important information based on client's criteria.
- Methods: clients can execute programs, etc. based on methods defined on the server.

### Platform Independence

Given the vast

range of available hardware platforms and operating systems, platform independence is essential. OPC UA functions on any of the following and more:

- Hardware platforms: traditional PC Hardware, cloud-based servers, PLCs, micro-controllers (ARM etc.)
- Operating Systems: Microsoft Windows, Apple OSX, Android or any distribution of Linux, etc.

OPC UA provides the necessary infrastructure for interoperability across the enterprise, from machine-to-machine, machine-to-enterprise and everything in-between.

### Security

Security is one of the most important things to consider when choosing which technology to use. OPC UA is firewall-friendly while addressing security concerns by providing a suite of controls:

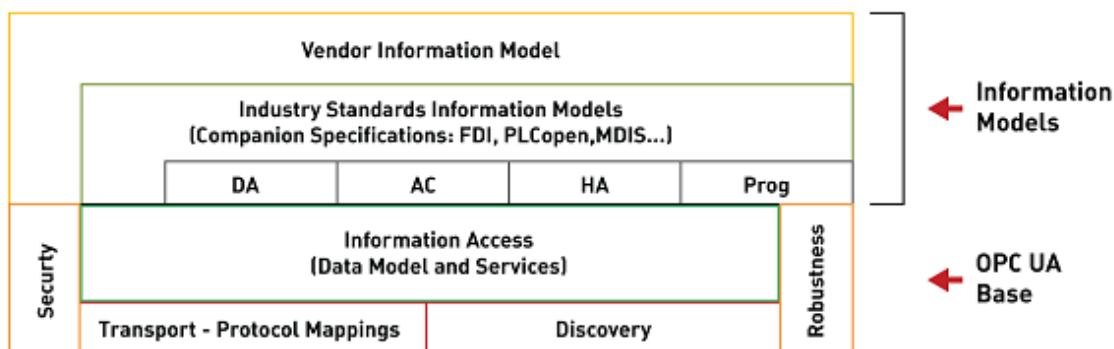
- Transport: numerous protocols are defined providing options such as the ultra-fast OPC-binary transport or the more universally compatible SOAP-HTTPS, for example.
- Session Encryption: messages are transmitted securely exactly at 128 or 256 bit encryption levels.
- Message Signing: messages are received exactly as they were sent.
- Sequenced Packets: exposure to message replay attacks is eliminated with sequencing
- Authentication: each UA client and server is identified through OpenSSL certificates providing control over which applications and systems are permitted to connect with each other.
- User Control: applications can require users to authenticate (login credentials, certificate, etc.) and can further restrict and enhance their capabilities with access rights and address-space 'views'.
- Auditing: activities by user and/or system are logged providing an access audit trail

### Extensible

The multi-layered architecture of OPC UA provides a 'future proof' framework. Innovative technologies and methodologies such as new transport protocols, security algorithms, encoding standards, or application-services can be incorporated into OPC UA while maintaining backwards compatibility for existing products. UA products built today will work with the products of tomorrow.

### Information Modelling

The OPC UA information modelling framework transforms data into information. With complete object-oriented capabilities, even the most complex multi-level structures can be modelled and extended. Data-types and structures are defined in profiles. For example, the existing OPC Classic specifications were modelled into UA profiles which can also be extended by other organizations. :



OPC UA Base Services Architecture

More info:  
<https://opcfoundation.org/about/opc-technologies/opc-ua/>

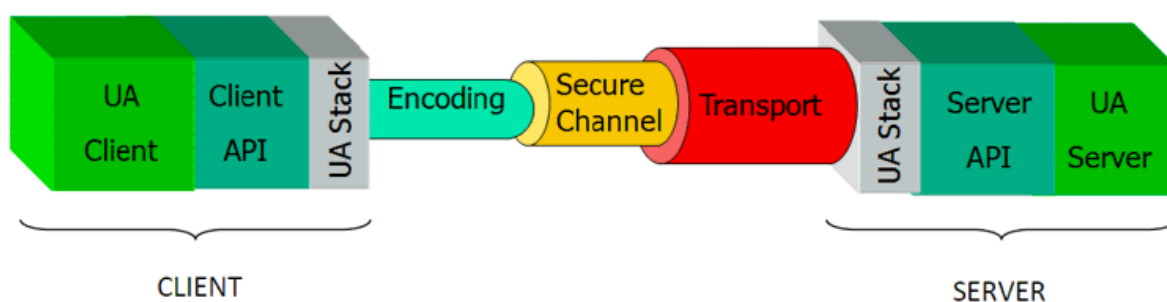
## Independent Data Transport

Transport is one of the essential components of OPC UA data modules. Transport is the infrastructure by means of which data is connected between Client and Server. In Movicon.NExT data are independent from the transport you wish to use or need to use for connections towards third party systems in OPC UA.

Different types of transport can be used according to whether data is to be encoded or protected.

The transports offered by the Data Server are described in the topics on the Movicon.NExT I/O Data Server.

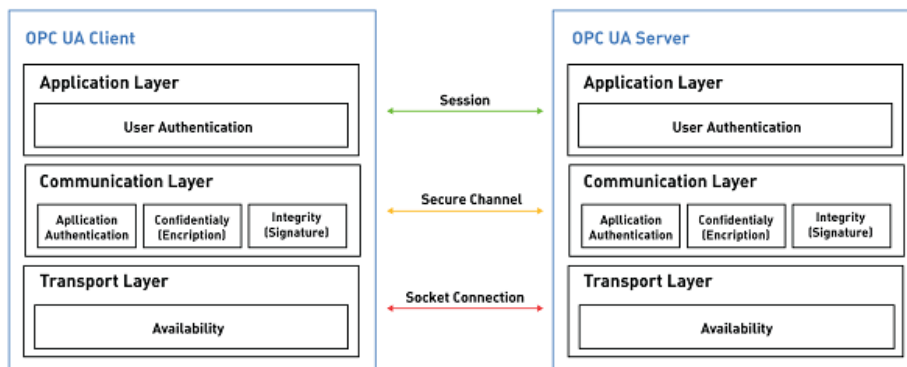
In order to obtain communication between OPC UA Server and Client, you should always make sure that the transport you wish to use is appropriate and consistent with the purposes you intend to use it for.



As OPC UA uses a client-server architecture, it is normal that an application assumes both these roles. This is due to the fact that often the Server side is implemented in physical devices (device to device communication). A typical OPC UA application is composed of three software layers as shown in the above image.

## 15.2. OPC UA Security

The OPC UA security architecture is a solution that consents the implementation of security functions in various parts and has a multi-layer structure: Application Layer, Communication Layer and Transport Layer.



The Application Layer is used to transmit information between Client and Server which have established an OPC-UA Session on a Secure Channel (which is found at the Communication Layer). The Secure Channel secures the data exchanged in a session. The Transport Layer is responsible for transmitting and receiving data.

The security in OPC UA is realized through the defining certain security mechanisms that consent the Application Authentication in a Secure Channel and User Authentication in the Session.

### Secure Channel definition

The Secure Channel, where the Application's Authentication takes place, ensures that data is exchanged safely in the following two modes:

- **Sign:** securing data integrity by using digital signatures. The security mechanism providing use of the digital signatures requires that:
  - Each side (Client and Server) possesses two keys (Public/Private Key)
  - Each side (Client and Server) makes their public key known
  - Each message exchanged between both sides is signed using the sender's private key and verified by the receiver with the sender's public key.
- **Encrypt:** securing the confidentiality of data by using asymmetric cryptography. The security mechanism used for asymmetric cryptography requires that:
  - Each side (Client and Server) possesses two keys (Public/Private Key)
  - Each side (Client and Server) makes their public key known
  - Encryption: Each message exchanged between both sides is encrypted using the public key of the counterpart and decrypted with each sides own private key.

The Secure Channel definition is created by applying a Security Profile and a Security Policy made available by the Server. The Security Profiles (also called Security Modes) are a set of three predefined security profiles:

- None: no security, channel is not safe.
- Sign: data integrity secured with digital signature (Application Certificate X.509). The digital signature is applied by the sender when exchanging data. The receiver can then verify whether the data actually comes from the expected sender.

- **Sign & Encrypt:** Data integrity and confidentiality secured with digital signature and encryption. First the sender's digital signature is applied to exchanged data which are then encrypted (Application Instance Certificate X.509).

The Security Policy indicates the length of the key and the algorithm used in the Security Profile; for example: Basic128Rsa15, Basic256, Basic256Sha256, Aes256-Sha256-RsaPss, etc...

The Security Policy and Security Profile combination forms the Security Level which indicates the degree of security of the Communication Level, where Level 0 is the lowest level. Clients can then choose the Security Policy and Security Profile simply by comparing them with the Security Levels.

### Session definition.

At Application Level however, the security mechanisms that consent user authentication (and as a consequence their authorization) ensure access for specific users (and therefore their roles) during the Session configuration. There are three different authentication modes at this level: anonymous, with credentials (username/password), with digital signature (User Certificate X.509)

### Secure Channel and Session establishment

The creation of a SecureChannel is principally based on the choice of Session Endpoint. Each OPC UA Server offers one or more Session Endpoints as follows:

- **Endpoint Url:** the endpoint's network address used by client to establish a SecureChannel
- **Server's Application Instance Certificate:** contains the Server's public key
- **Security Policy:** key length and algorithm used in the Security Mode
- **Security Mode:** none, Sign or Sign&Encrypt
- **User Authentication:** anonymous, username/password, User Certificate X.509
- **Transport Protocol:** specifies the stack characteristics used by the EndPoint: encoding, security, transport.

In order to establish a Session in a secure connection between an OPC UA Client and a OPC UA Server you need to carry out this steps:

1. **First step:** choose a Session Endpoint. The client selects a Session Endpoint and proceeds with validating the Server's Application Instance Certificate. If the certificate is trustworthy, the client can go ahead with the next steps.
2. The Client sends an Open Secure Channel request in accordance to the selected Session Endpoint Security Mode.
  - When Security Mode level is "None", the request is sent without using any security mechanisms.
  - When Security Mode level is "Sign", the request is sent using the Private key associated to Client's Application Instance Certificate as the signature.
  - When the Security Mode level is "Sign&Encrypt", the request is sent using the Private key associated to Client's Application Instance Certificate as the signature along with the code using the Public Key of the Server's Application Instance Certificate (in addition to the signature).
3. The Server receives the Open Secure Channel message and validates the Client's Application Instance Certificate contained in the message. When the certificate returns valid, the Server proceeds with interpreting the request in accordance with the selected Security Policy and Security Mode.
  - When the Security Mode level is "None", the request is received without applying any security mechanisms.
  - When the Security Mode level is "Sign", the request's signature is verified using the Public Key associated to the Client's Application Instance Certificate.

- When the Security Mode level is "Sign&Encrypt", the request is verified using the Public Key associated to the Client's Application Instance Certificate and then decrypted using the Private Key associated to the Server's Application Instance Certificate del Server

When the Server accepts, it sends the response to the Client using the same method used for the request and a Secure Channel is then established.

4. Once the Secure Channel has been established, the Client proceeds with sending a Create Session request to the Server. Once created it then has to be activated by passing the User Credentials to the Server.

## Appendix: asymmetric encryption and digital signatures

To respond to different needs, messages between two interlocutors can be encrypted or signed (or signed and then encrypted).

**The asymmetric encryption**, also known as a pair of cryptographic keys or a cryptographic public key is a type of encryption where messages are encoded by the agent who possesses a pair of keys:

- the 'private key', which is personal and secret, is used for decoding an encrypted document;
- the 'public key', which must be distributed, is used to encrypt a document destined to the person who possesses the relative private key.

In a cryptographic public key system, anyone can encrypt a message using the receiver's public key, but this message can only be decrypted with the receiver's private key.

The basic idea of asymmetric encryption becomes clearer when using a postal analogy where:

- the sender is Alice
- the receiver is Bob
- the locks are the public keys
- the keys to the locks are the private keys.

Communication between Alice and Bob takes place according to these steps:

1. Alice asks Bob to send him her padlock already opened. However the key to the lock is jealously guarded by Bob.
2. Alice receives the padlock and uses it to close the parcel then she sends it to Bob.
3. When Bob receives the parcel, he can open it with the key of which he is the only owner.
4. On the other hand, if Bob wants to send a parcel to Alice, he must ask her for the padlock so that he can use it to close the parcel. Once Alice receives the parcel, she can open it with her private key of which she is the only owner.

As you can see, in order to lock the parcel you need the receiver's padlock (public key) and to decrypt or in other words open the padlock you need to use the key of the padlock's owner (private key). This is how the asymmetrical encryption/decryption process works.

The digital signature is usually identified as the technology with which a document's authenticity is certified or data signed by who issued it. The digital signature system requires that the sender who issued the message uses their private key to generate information in association with the message to certify its origin.

It is important to note how both the encryption and the signature are based on the exchange of public keys. When the message is encrypted, the public key used is the one belonging to the receiver. When the message is signed, it is the sender's key that is used. In both cases, the value of the public keys is not confidential and the critical part lies in guaranteeing their authenticity. It must therefore be made certain that a specific public key effectively belongs to the interlocutor for whose message needs to be encrypted or whose signature needs to be verified. If a third party discovers the receiver's public key and replaces it with their own, the encrypted message contents will be unveiled and it will not be possible to verify the validation of a digital signature.

The distribution of public keys is a crucial problem in public key technology and is resolved by deploying electronic certificates. The public key certificates (X.509) are a reliable and secure tool with which public keys can be distributed and made known to end users while guaranteeing data authenticity and integrity.

The structure of an X.509 certificate is as follows:

- Version Number
- Serial Number
- Signature Algorithm ID: encryption algorithm used
- Issuer Name: identifies the Certificate Authority (CA) who issued and signed the certificate
- Valid from/to: validity period
- Public Key: Certificate's public key
- Signature: digital signature created by issuer. The digital signature imposed by using the issuer's Private Key.

Each X.509 certificated is associated with a Private Key.

## 15.3. OPC UA Certificates

Movicon.NExT uses public key certificates (X.509) as a reliable and safe tool through which public keys are exchanged with guaranteed authenticity and integrity within an OPC UA communication context. These certificates are generated automatically during the setup phase and include the following:

Certificate File	Application/Service
"Platform.NExT.v2 Web Server [thumbprint].der"	WebHMI
"Platform.NExT.v2 Scheduler [thumbprint].der"	Scheduler Server
"Platform.NExT.v2 AlarmDispatcher [thumbprint].der"	Alarm Dispatcher
"Platform.NExT.v2 IO Server [thumbprint].der"	I/O Data Server
"Platform.NExT.v2 MoviconNExT [thumbprint].der"	OPC UA Browser /Client Movicon.NExT

and are found at this location: "%ProgramData%\OPC Foundation\CertificateStores\MachineDefault\certs\"

Movicon.NExT, in addition to the certificate repository of its applications/services, provides another two folder files:

- Trusted Certificates: "%ProgramData%\OPC Foundation\CertificateStores\UA Applications\certs"
- Rejected Certificates: "%ProgramData%\OPC Foundation\RejectedCertificates\certs"

### How to automatically accept untrusted certificates

Untrusted certificates can be accepted automatically, and therefore inserted in the Trusted Certificates folder, by using configuration options provided.

To make that, for example, the I/O Data Server automatically accept third party OPC UA Client certificates, simply set the **<AutoAcceptUntrustedCertificates>true</AutoAcceptUntrustedCertificates>** option in the "UFUAServer.UAServer.Config" files found in the Movicon.NExT installation folder.

To make the Client or OPC UA Browser automatically accept third party OPC UA Server certificates, simply set the **<AutoAcceptUntrustedCertificates>true</AutoAcceptUntrustedCertificates>**

option in the "OPCUAClient.Config" file found in the Movicon.NExT installation folder.

### **Movicon.NExT Client <--> Movicon.NExT I/O Data Server (Users Management)**

When using the Movicon.NExT Users Management, data exchanged between Client and Server takes place with the "Sign" security policy by means of using certificates.

In this condition, if the the project has been configured with distributed Client and Server architecture, the exchange of security certificates between Server and Client will be needed:

This is established by copying the "Platform.NExT.v2 IOserver [thumbprint].der" file of the PC where the I/O Data Server is running in the Trusted Certificates folder of the PC where the Movicon.NExT Client is running.

### **Movicon.NExT I/O Data Server <--> third party OPC UA Client**

This example shows how to exchange certificates manually to put third party OPC UA Client in communication with Movicon.NExT I/O Data Server. In this case simply:

- copy the "Platform.NExT.v2 IOserver [thumbprint].der" file in the third party OPC UA Trusted Certificates folder.
- copy the third party OPC UA Client's certificate file in Movicon.NExT's Trusted Certificates folder.

### **Movicon.NExT OPC UA Client Driver <--> Third Party OPC UA Server**

This example shows you how to manually exchange certificates to put the Movicon.NExT I/O Data Server in communication with a third party OPC UA Server by means of using OPC UA Client communication driver, simply:

- copy the "Platform.NExT.v2 IOserver [thumbprint].der" file in the third party OPC UA Server's Trusted certificates folder.  
When importing Tags in the project using OPC UA, it will be necessary to also copy the "Platform.NExT.v2 MoviconNExT [thumbprint].der" certificate in the third party OPC UA Server as indicated in the next paragraph.
- copy the third party OPC UA Server certificates file in Movicon.NExT's Trusted Certificates folder.



The OPC UA Server might automatically insert the "Platform.NExT.v2 IOserver [thumbprint].der" certificate in the list of Rejected certificates. In this case, simply move it to the list of Trusted certificates.

### **Movicon.NExT Client/OPC UA Browser <--> Third Party OPC UA Server**

This example will show you how to exchange certificates manually for putting the Movicon.NExT Client or OPC UA Browser in communication with a third party OPC UA Server, simply

- copy the "Platform.NExT.v2 MoviconNExT [thumbprint].der" file third party OPC UA Server's Trusted certificates folder.
- copy the third party OPC UA Server certificates file in Movicon.NExT's Trusted Certificates folder.

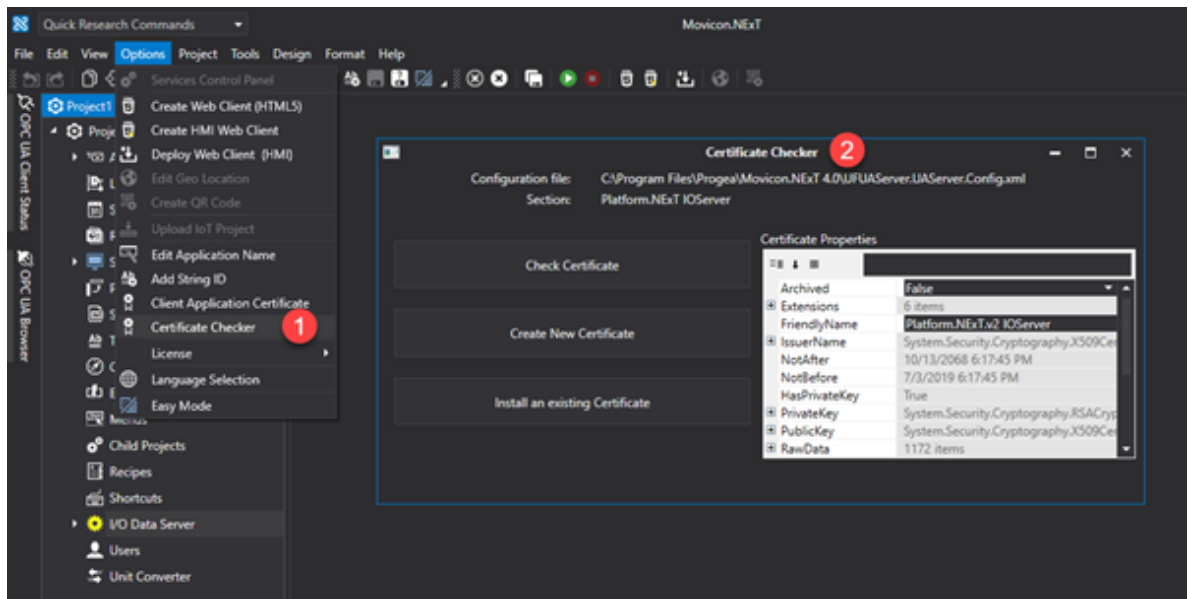


The OPC UA Server might automatically insert the "Platform.NExT.v2 IOserver [thumbprint].der" certificate in the list of Rejected certificates. In this case, simply move it to the list of Trusted certificates.

### **Client/OPC UA Browser Certificates Management**

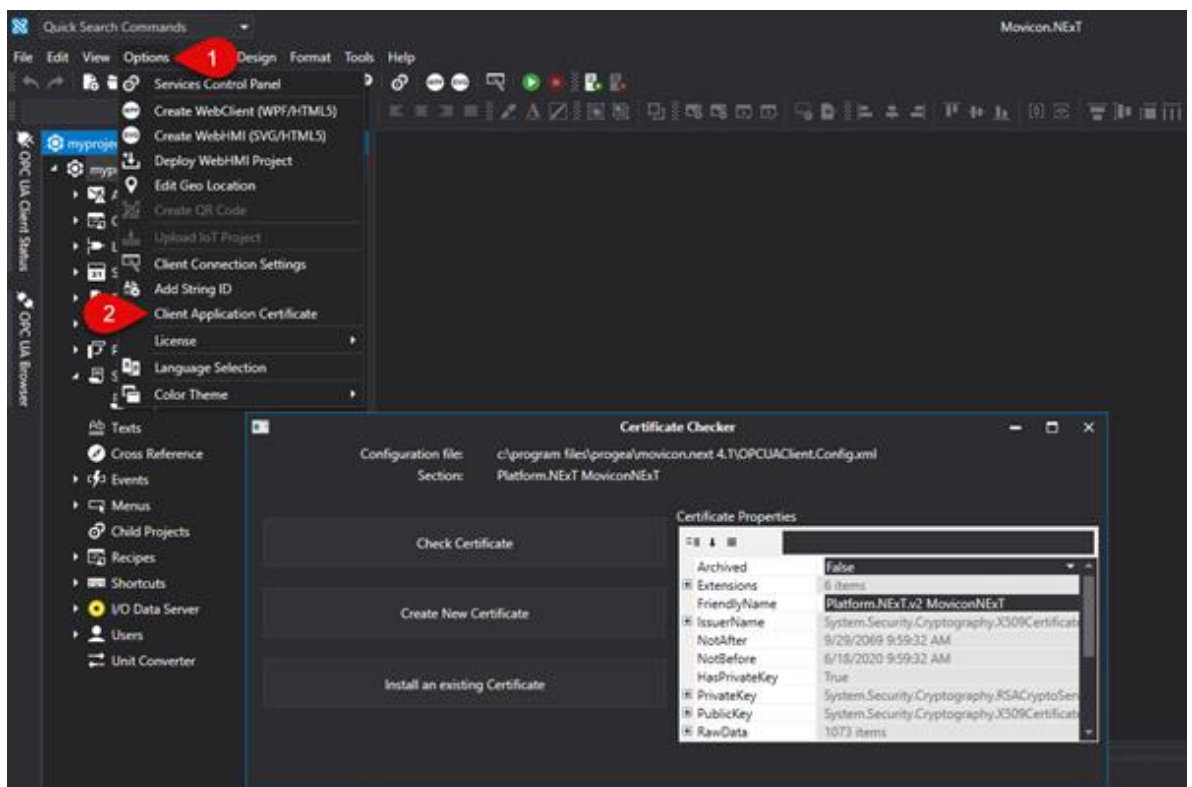
The certificate of the Movicon.NExT Server side called "Platform.NExT.v2 IOserver" may be checked, renewed or replaced using the configuration tool which is accessible (1) by expanding the I/O Data Server resource, (2) and selecting the Settings item and then (3) Selecting the Certificate Checker.





### I/O Data Server Certificate Folder

The certificated of Movicon.NExT Client side called "Platform.NExT.v2 MoviconNExT" can be checked, renamed or replace by using the configuration tool accessible from the menu > option and selecting the Application Certificate (2) item.



## 15.4. OPC UA Server

### 15.4.1. OPC UA Server

The Movicon.NExT Servers conforms to the OPC UA specification and consents connectivity to any other third+ party OPC UA Client platform or device supporting these specifications:

- DA (Data Access)
- AC (Alarms and Conditions)
- HA (Historical Access)



**The OPC UA Server functionality is optional and needs to be enabled on your software product license.**

### Server Certification

Movicon.NExT's OPC UA Server has been successfully certified by the OPC Foundation's OPC Certification Test Lab. The Test Lab carried out validation tests to verify full conformity with the standard (rel. 1.02) and Performance Stress and Load.

Movicon.NExT obtained Certification from the OPC Foundation to ensure users with guaranteed:

- Compatibility
- Interoperability
- Robustness
- Usability
- Efficiency



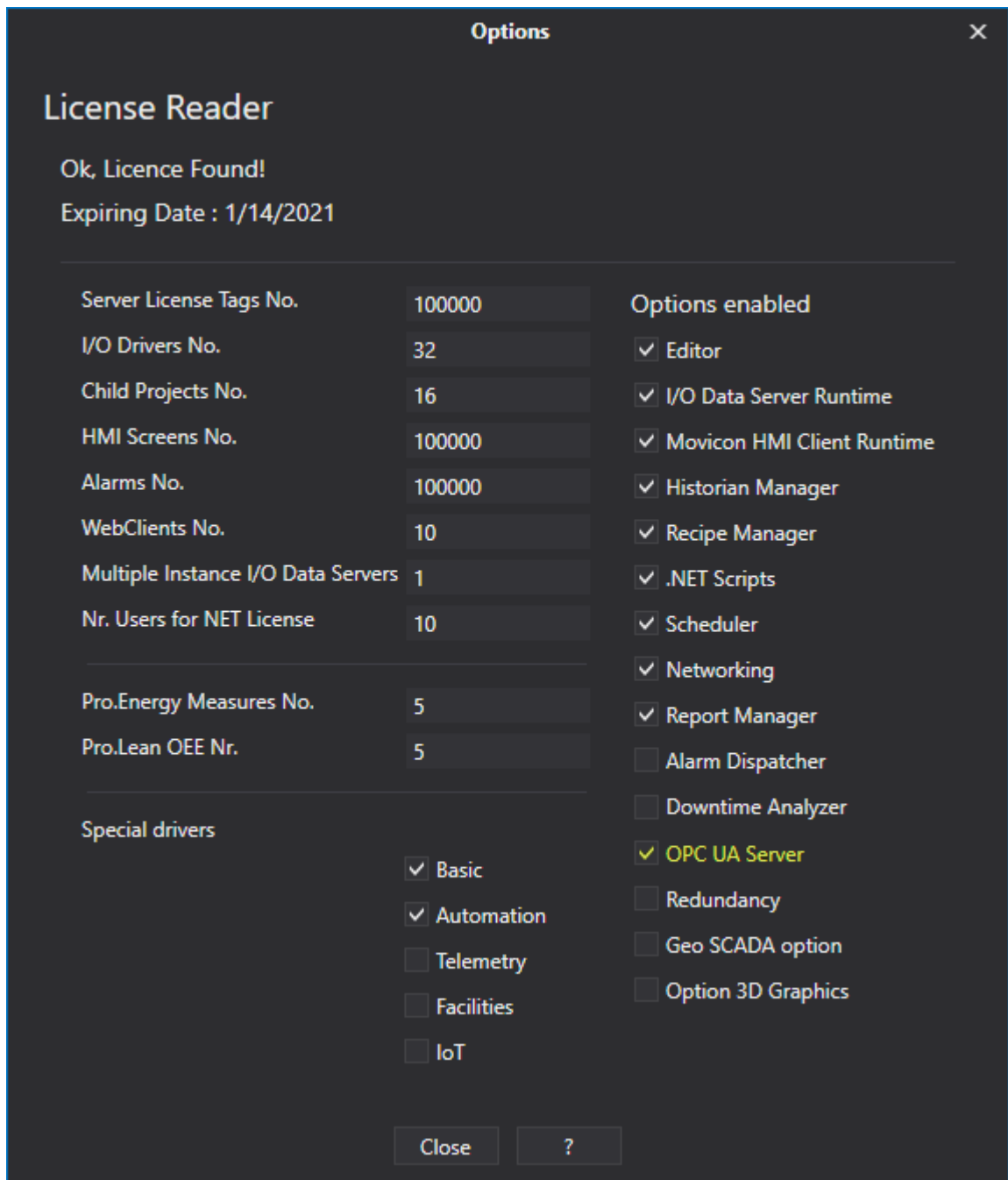
#### 1. Certificate

**Number: 1506CS006A**

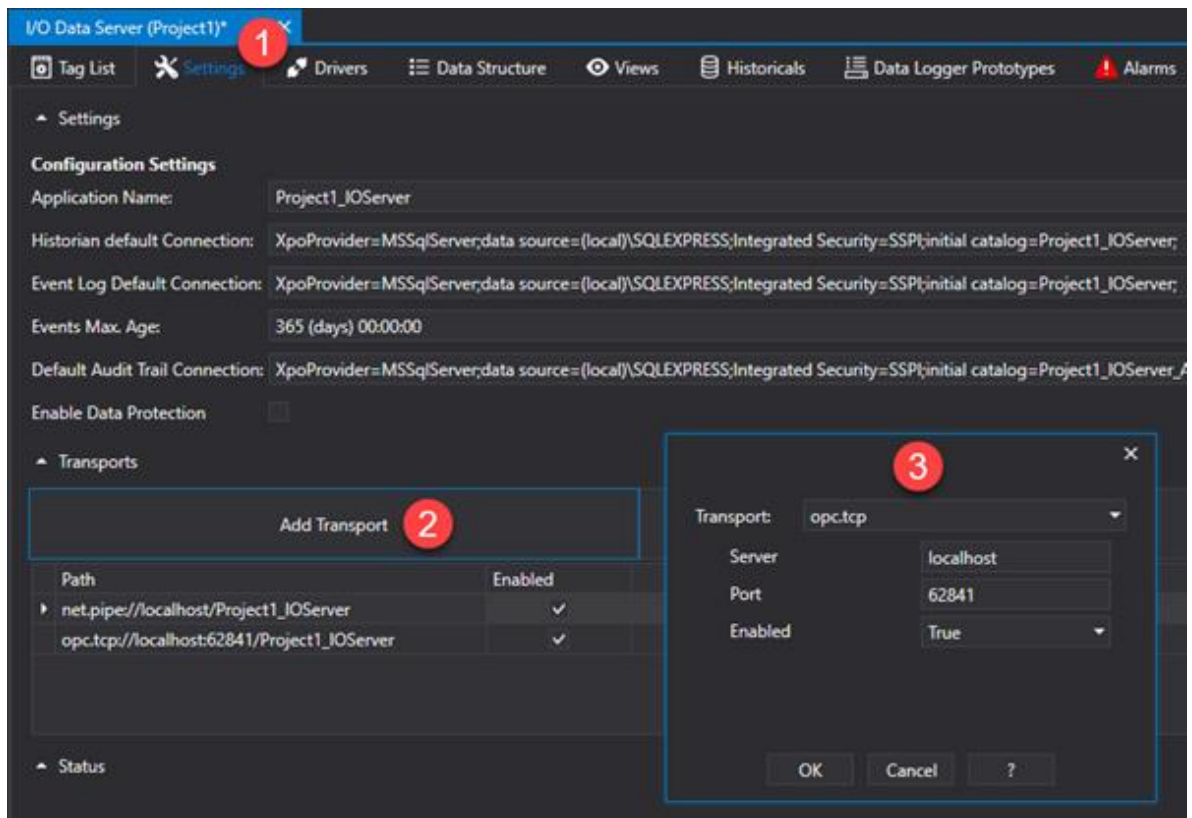
### Third party OPC UA Client Access

In order for the Movicon.NExT's I/O Data Server Tags to be visible and connected by means of using a third party OPC UA Client browser, the following points must be taken into consideration:

1. The OPC UA Server must be enabled on the Server's Runtime license



2. A network transport type must be configured in the I/O Data Server Settings that is adaptable for OPC UA communications such as tcp, http or https. As shown in the screenshot below, open the I/O Data Server Settings tab (1) and select Add Transport (2) in the transport definition area. In the network transport definition dialog field (3) select the transport desired and configure its properties. We recommend that you keep the 'localhost' as Server so that the hostname defined by the operating system is used.



The OPC Foundation's OPC UA Local Discovery Server (LDS) gets installed with the Movicon.NExT Setup. This Server provides the infrastructure needed to expose OPC UA Servers; which have been started up in a machine, to make them visible to OPC UA Clients. If any problems with connecting or browsing between Client and Server occur, you should verify whether the OPC UA Local Discovery Server (opcualds.exe) has been started up and is working correctly (this process is found in the Windows Task Manager list headed UALDS).

## Session Endpoint, Security Mode and Certificates

Movicon.NExT's I/O Data Server exposes three Endpoints with the following Application Authentication options:

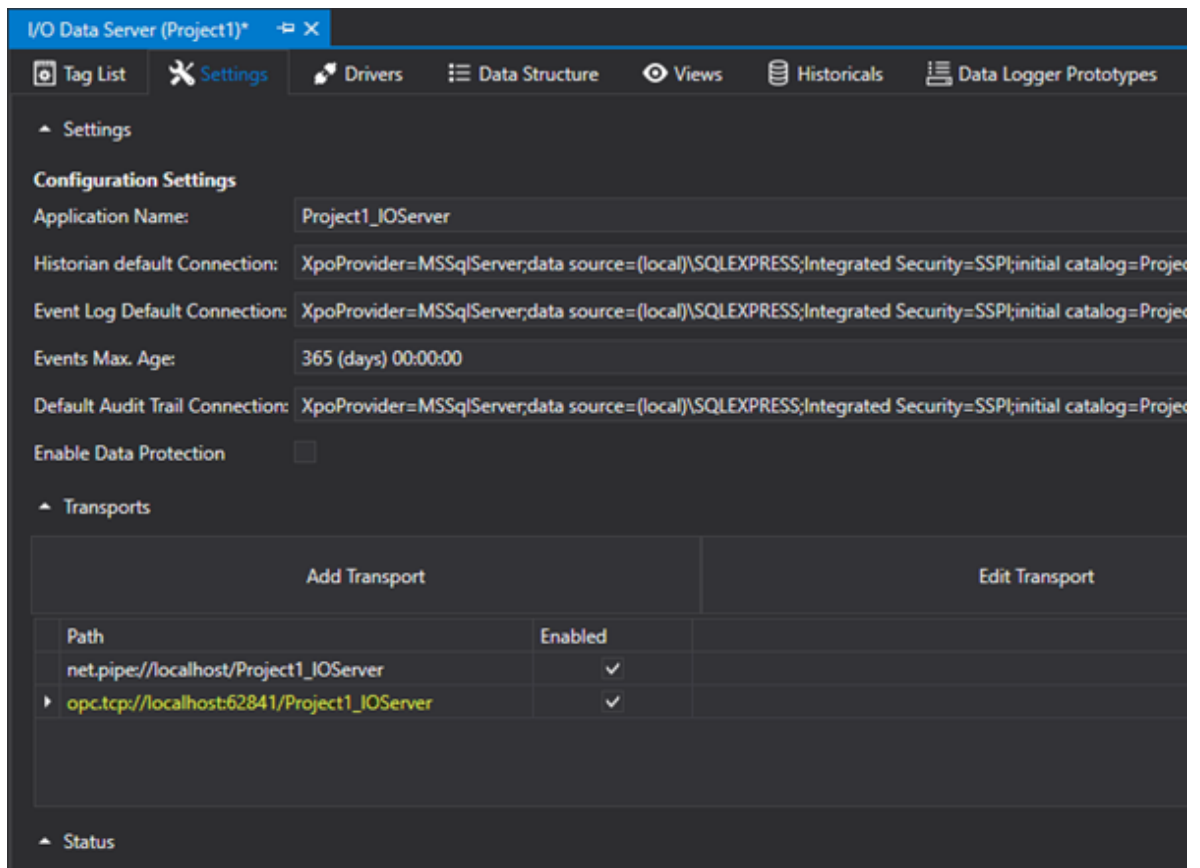
Security Mode	Security Policy	Security Level
None	None	Level0
Sign	Basic256Sha256	Level1
Sign&Encrypt	Basic256Sha256	Level5

All authentication types defined by the OPC UA standard are supported for user authentication:

Anonymous, Username/Password, X.509 User Certificates

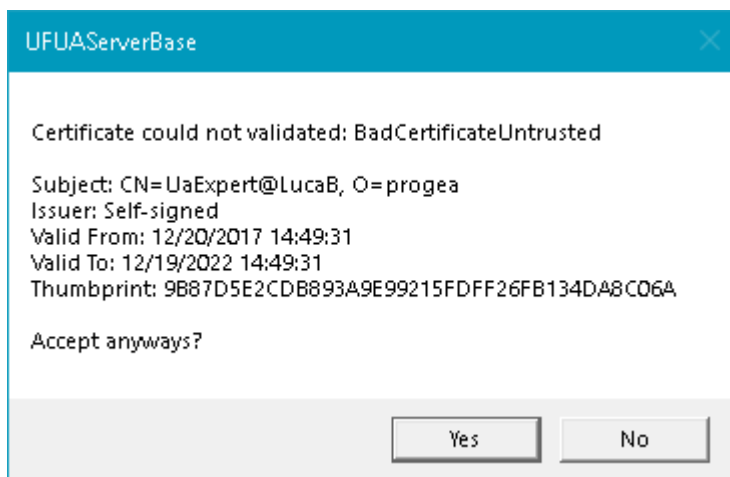
These options can be changed by means of editing "UFUAServer.UAServer.Config" file which resides in the Movicon.NExT installation folder.

The Session Endpoint to use in the OPC UA Client to activate a Secure Channel with the Server is one of those defined in the list of Transports in the I/O Data Server Settings.



*In reference to the previous screenshot, the session url to use would be:  
opc.tcp://<hostname\_del\_server>:62841*

When the Security Mode is Sign or SignAndEncrypt type, the Channel will open after the Application Instance Certificates have been exchanged. The Movicon.NEXt Server automatically sends its OPC UA certificate to the OPC UA Client and receives and accepts the OPC UA Client certificate interactively by proposing a dialog as shown below:



**The Client certificate confirm dialog window is only available when the I/O Data Server has not been started up as service. However, it is possible to make the Server automatically accept Untrusted certificates by setting the `<AutoAcceptUntrustedCertificates>true</AutoAcceptUntrustedCertificates>` option in the**

"UFUAServer.UAServer.Config" file residing in the Movicon.NExT installation folder.



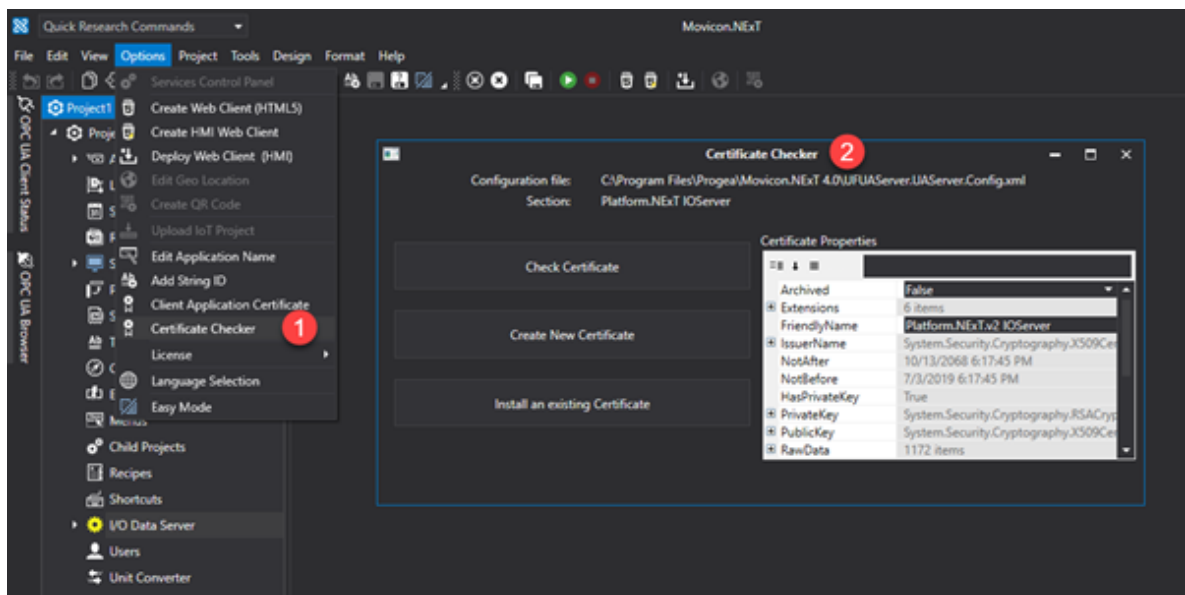
If you wish to make the OPC UA Client Certificate permanently Trusted on the Server, you will need to copy the Client's Certificate to the directory "%ProgramData%\OPC Foundation\CertificateStores\UA Applications\certs" directory.



If you need to provide the I/O Data Server's Application Instance Certificate to the OPC UA Client, you will find it in the file called "Platform.NExT.v2 IOserver" from the "%ProgramData%\OPC Foundation\CertificateStores\MachineDefault\certs\" folder

## Certificate Management

The Movicon.NExT Server's certificate called "Platform.NExT.v2 IOserver" can be controlled, renewed or replaced using the configuration tool which is accessed by expanding the I/O Data Server (1) and selecting the Settings tab (2) and then selecting the Certificate Checker (3).



The Client certificate confirm dialog window is only available when the I/O Data Server has not been started up as service.

For further information on managing OPC UA certificated, please see "OPC UA Certificates "

## 15.5. OPC UA Client

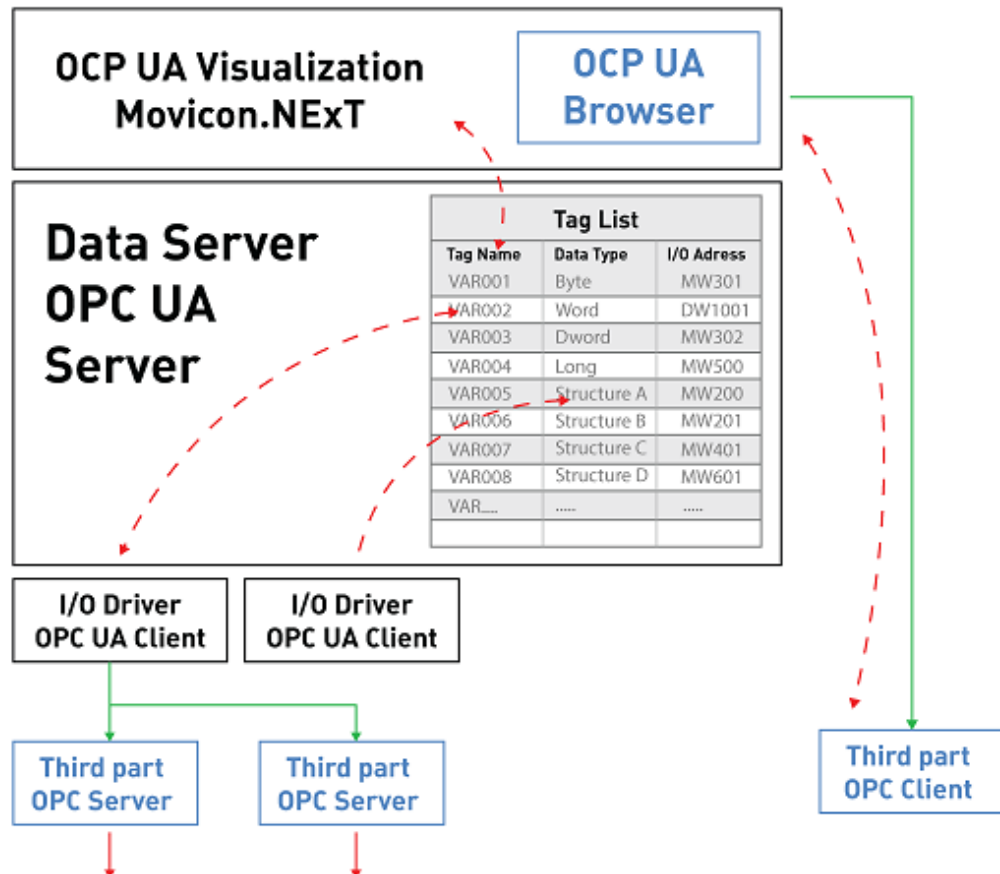
### 15.5.1. OPC UA Client

The Movicon.NExT client conforms to the OPC UA specifications and consents connectivity towards any other third party OPC UA Server platform or device that supports these specifications:

- DA (Data Access)
- AC (Alarms and Conditions)
- HA (Historical Access)

### Third party OPC UA Server connectivity

Client-Server connectivity according to the OPC UA Standard can occur on the Movicon.NExT platform's Client side and Server side as needed.



**The integrity and confidentiality of data exchanged with third party OPC UA Servers secured by the exchanging of Application Certificates in the Application Authentication phase (Sign or SignAndEncrypt). Authentication by means of using X.509 Certificates is not supported at User Authentication level which continues to use the Anonymous and Username/Password modalities.**

### Client side connection using OPC UA Browser

In the visualization client it is possible to associate symbols and graphical objects on screen to Tags in the Server's Address Space as well as connect them directly to a third party OPC UA Server using the OPC UA Browser.

This modality lets you associate symbols or graphical objects to Tags which can be selected by browsing OPC UA Server items live.

In this way, the 'Client only' architecture can also be used by involving the platform's graphical part only: Movicon.NExT visualization client.

### Server side connection using the OPC UA Client Driver

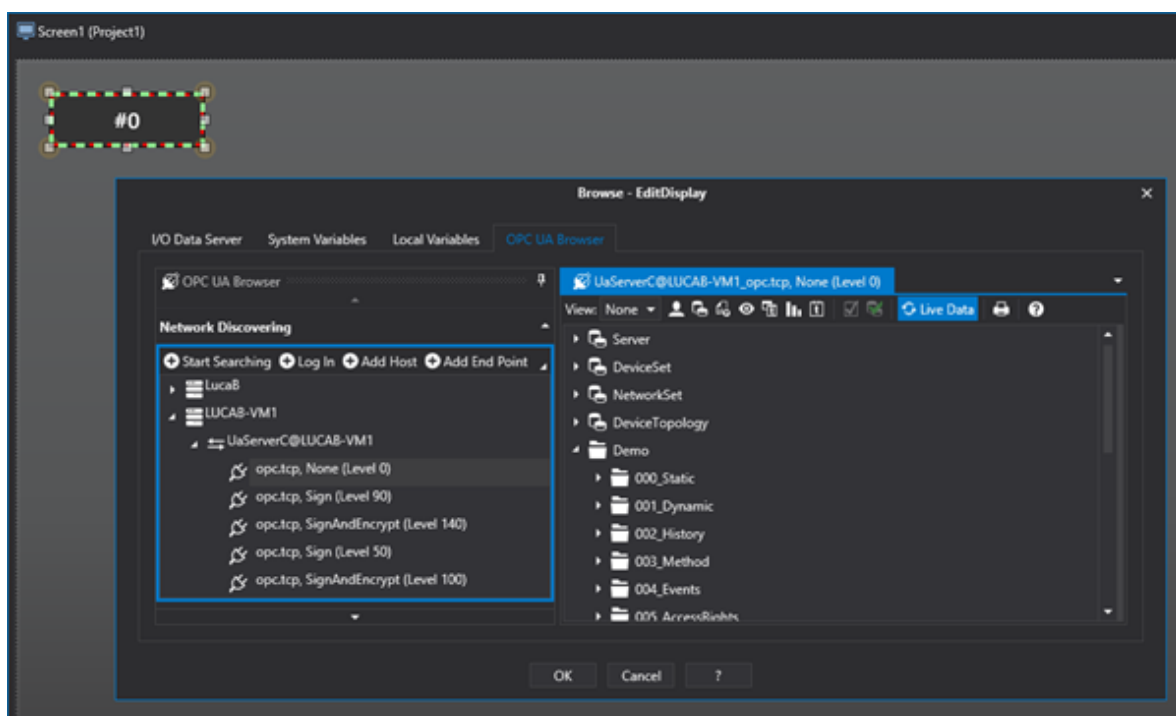
In order to make data deriving from other OPC UA Servers available to others by means of the Movicon.NExT Server, you will need to use the OPC UA Client communication driver provided for this purpose.

In this way, data will be available by means of Tags defined/imported in the platform's Address Space and therefore available to the entire system.

### 15.5.2. Client-side connection via OPC UA browser

It is possible to associate directly, from any object or display symbol or command insertable in the synoptics, an item displayed by any OPC UA Server compatible. In this way, the graphic element will be associated directly with the Item of the OPC UA Server, and not to a project Tag.

To achieve this, simply use the video selection window Tags, and from here select the Tab corresponding to the OPC UA browser.

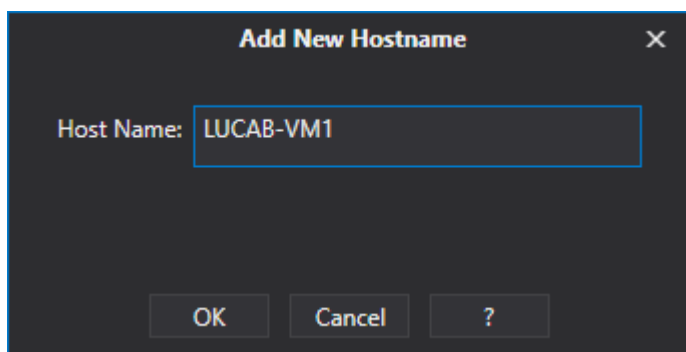


### Endpoint connection

To end the procedure used to select an item from the OPC UA Server using the selection window, you will need to establish a Secure Channel by means of activating an Endpoint Session.

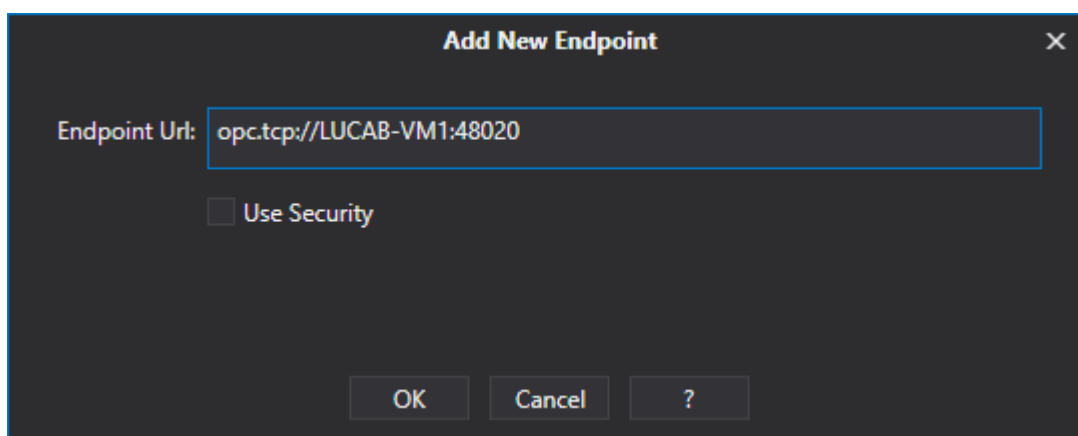
The Endpoint can be selected using one of three commands: "Start Searching", "Add Host" or "Add Endpoint".

A Host should be added if you do not know the Endpoints made available by the Server.





Otherwise, if you already know the Endpoint's url, you can select "Add Endpoint" to define the url using this format: "<Transport Protocol>://<hostname>:<port>", for example, "opc.tcp://server1:48020".



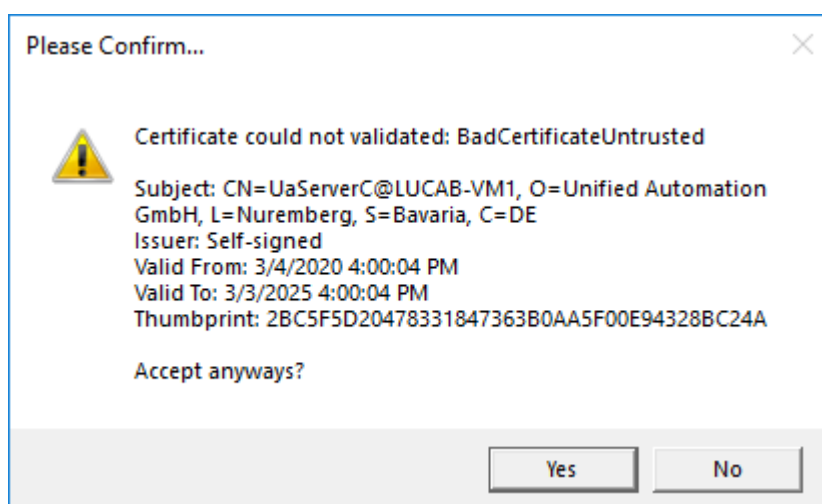
When you select the "Use Security" option, the Endpoint with a security level that is higher than those made available by the Server with a Sign or SignAndEncrypt Security Mode, will be selected.



**It would be preferable to use a hostname instead of an IP address to manually define the Host and Endpoint especially in those cases where the selected Endpoint uses the Sign or SingAndEncrypt Security Modes. Creating a Secure Channel, in these cases, involves the exchanging of public certificates between the Client and Server. The necessity to use a hostname (or IP address) depends on how the OPC UA Server's security certificate to be connected to has been defined.**

After having added an Endpoint or a Host that exposes several Endpoints, you will need to select one from the tree structure on the left with a double click to activate the communication channel towards the OPC UA Server.

If the security mode is Sign or SignAndEncrypt type, the Channel will open after the Application Instance Certificates have been exchanged. The Movicon.NExT OPC UA Browser receives and accepts the OPC UA Server certificate interactively by displaying a dialog window as shown in the screenshot below:



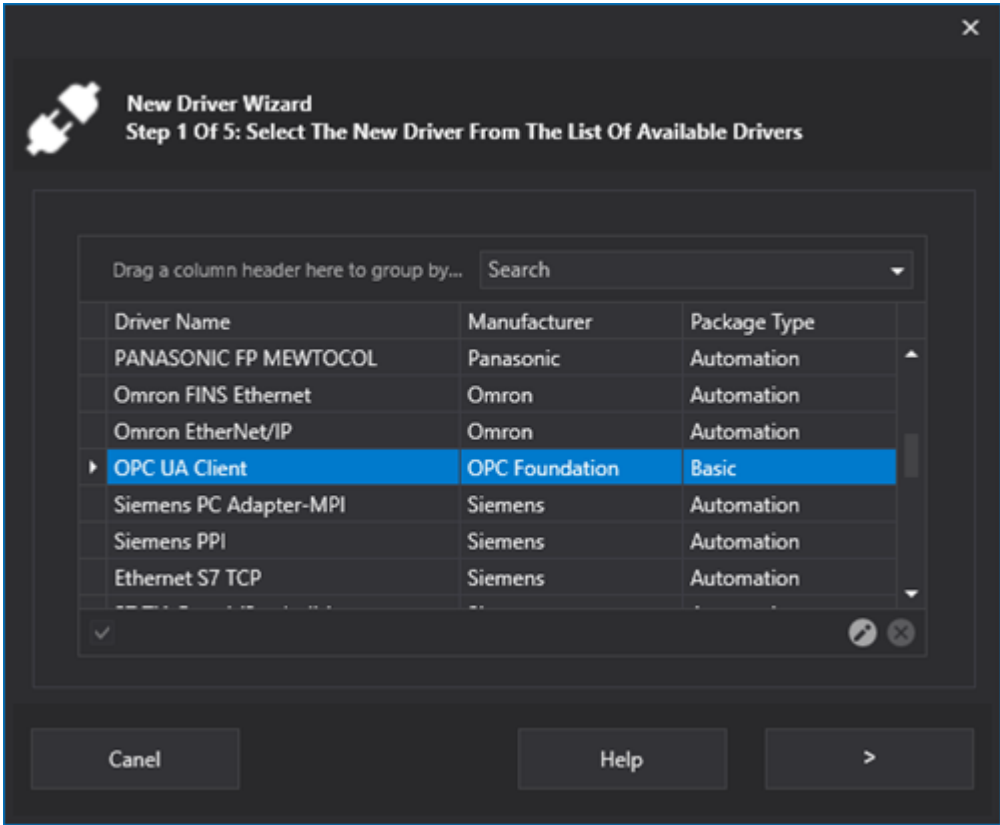
In this way the Movicon.NExT development environment, acting as Client, will accept the validity of the certificate for whole duration of the session.



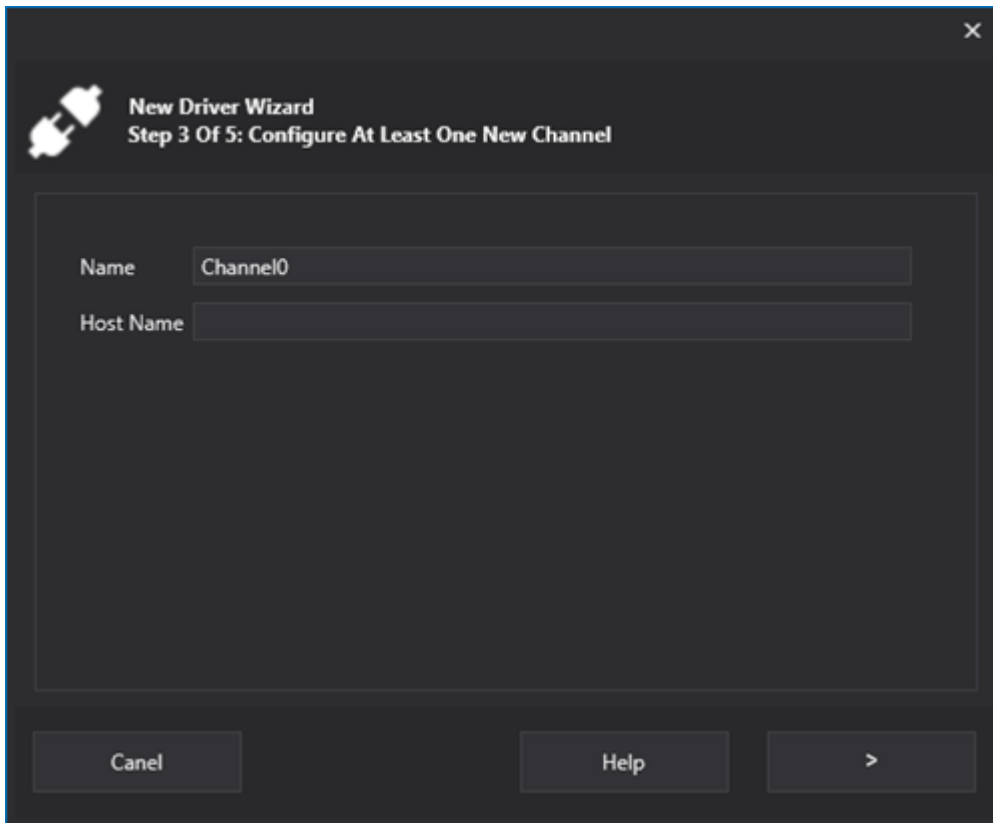
For further information on managing OPC UA certificates, please see "OPC UA Certificates"

15.5.3. Connecting to Server with an OPC UA Client communication driver

Connectivity towards an OPC UA server is usually accomplished by using an OPC UA Client communication driver.  
In this way the Movicon.NExT Server becomes Client of a second OPC UA Server.  
By means of using the Client OPC UA driver, the selected items will be available as Tags in the project's I/O Data Server's Address Space.



In addition to defining the name of the Channel, you must also define a Hostname to be used when connecting to the items. When tags are imported with the OPC UA Browser, this Hostname will be used to replace the one defined in the item's url.

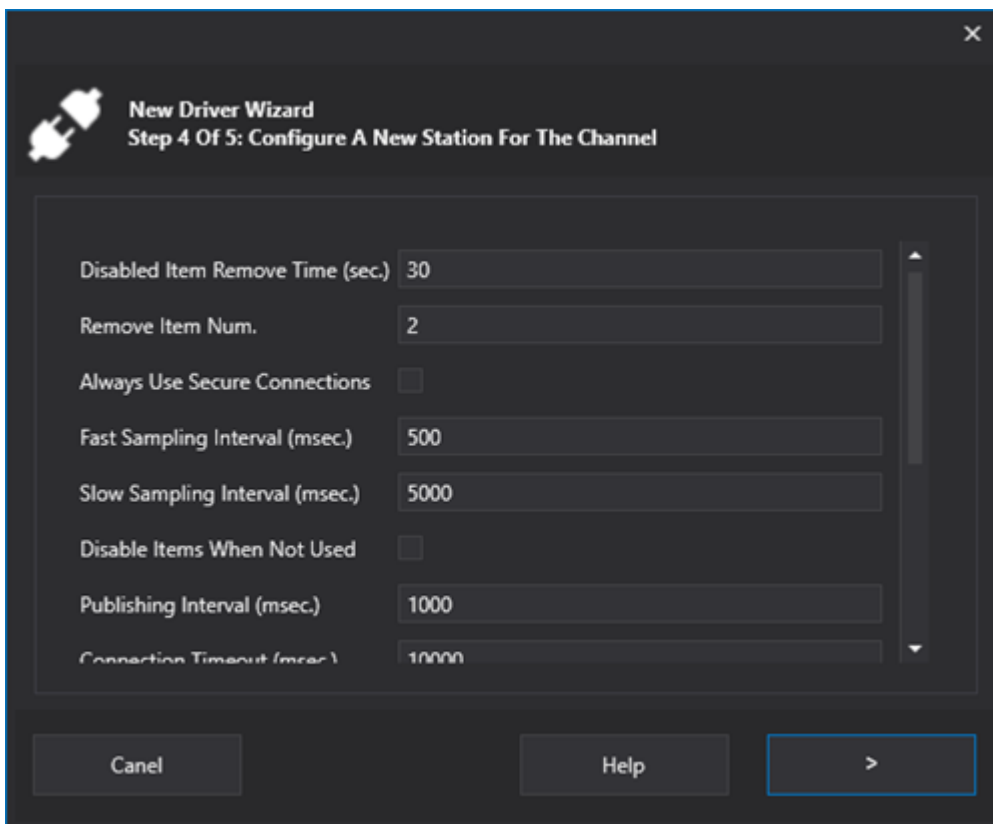


**New Driver Wizard**  
Step 3 Of 5: Configure At Least One New Channel

Name

Host Name

The following parameters are available for defining the Station:



**New Driver Wizard**  
Step 4 Of 5: Configure A New Station For The Channel

Disabled Item Remove Time (sec.)

Remove Item Num.

Always Use Secure Connections ☐

Fast Sampling Interval (msec.)

Slow Sampling Interval (msec.)

Disable Items When Not Used ☐

Publishing Interval (msec.)

Connection Timeout (msec.)

Disabled Item Remove Time (Sec.): this is used to set a delay time in seconds with which the inactive OPC UA subscriptions will be removed.

- Remove Item Num.: this is used to set the number of inactive OPC UA items to be removed at each time interval.

- Always Use Secure Connections: this is used to select the Endpoint with a Security Level higher than those made available by the Server with a Sign or SignAndEncrypt Security Mode type.

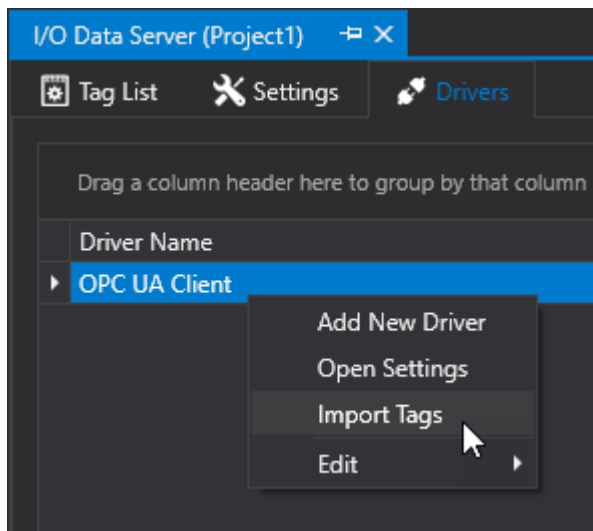


**The certificate will need to be exchanged with public keys when using the Security Sign or SignAndEncrypt Security Modes. Therefore, the OPC UA Server must be provided with the certificate used by the OPC UA Client called "Platform.NExT.v2 IOserver", which is available in the "%ProgramData%\OPCFoundation\CertificateStores\MachineDefault\certs\" folder. While the OPC UA Server's ".der" certificate must be copied to the "%ProgramData%\OPC Foundation\CertificateStores\UA Applications\certs\" folder.**

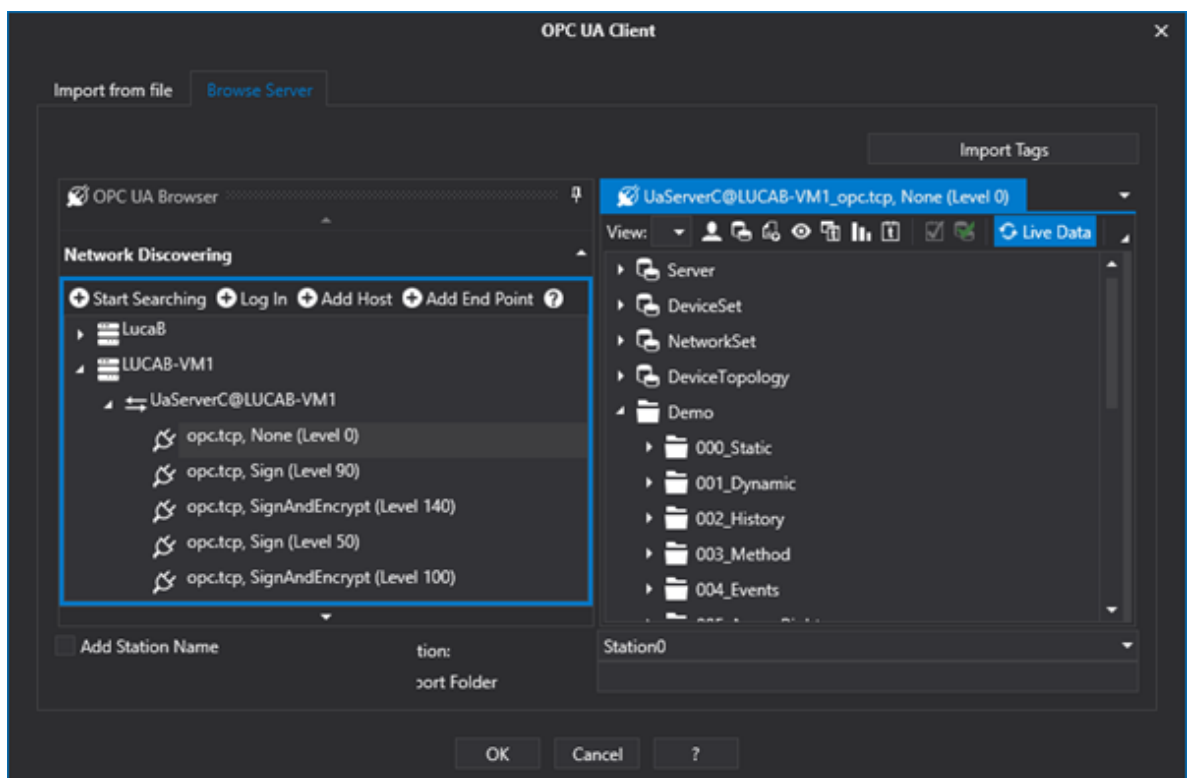
- Fast Sampling Interval: Defines the frequency with which existing and not-in-use tags are updated.
- Slow Sampling Interval: Defines the frequency with which existing tags that are going out-of-use are updated.
- Disable items When Not Used: Sets Tags as "Inactive" when not in use.
- Publishing Interval: Time in seconds of tag notifications towards Server.
- Connection Timeout: Connection timeout to Server.
- Username: Username to use for OPC UA Authentication at user level.
- Password: Password to be used for OPC UA Authentication at user level.
- Name: Name of Station.
- Channel: Channel that Station refers to.
- State/Command Variable: this property is used to assign a name to a supervision numeric variable (Byte type recommended) in order to control the communication status of the selected channel.

Bit 0 (State)	Connection Channel 0= connected 1= not connected
Bit 1 (State)	Primary Host Error State 0=Active 1=Inactive
Bit 2 (State)	Backup Host Error State 0=Active 1=Inactive
Bit 3 (State)	Connected Host 0=Active 1=Inactive

Once the configuration of the OPC UA Client driver has been defined, it is possible to import the Tags from the OPC UA Server by selecting Import Tags from the contextual menu:



Tag import can be done using the "Import from file" or "Browse Server" which used the OPC UA Browser as seen previously.



For further information on managing OPC UA certificates, please see "OPC UA Certificates"



# 16. Movicon WebHMI

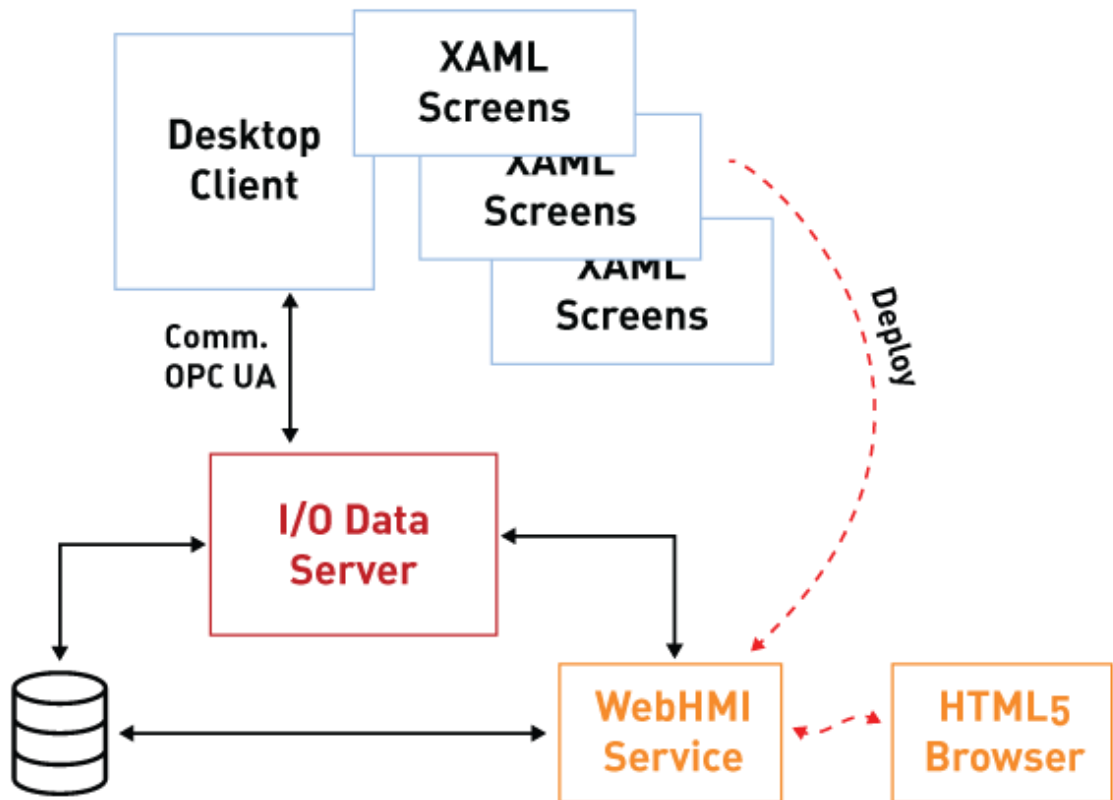
## 16.1. Introduction to using the WebHMI

WebHMI is a solution to use browsers to visualize HMI projects created with Movicon.NExT.

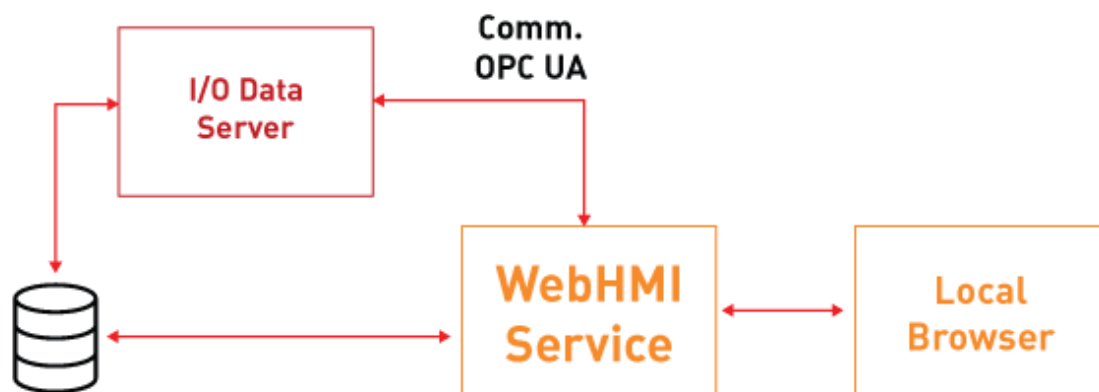
The new WebHMI Web Client makes project pages available in SVG/HTML5 format using an integrated Web Server which interfaced with the Movicon.NExT I/O Data Server through OPC UA.

The two main fields of application can be distinguished as follows:

1. WebHMI Client :To expose projects created with Movicon.NExT for access via HTML5 Browsers as an alternative to WPF/HTML5 WebClient.



2. WebHMI Runtime: As stand-alone applications on the HMI (Windows or Linux) that use the I/O Data Server and displays graphical pages locally using a browser opened at full screen.



In respect to the previous WPF/HTML5 Web Client solution, the WebHMI which could also be called SVG/HTML5 has distinct characteristics as described below.

### Using SVG graphics

The vector graphics used by Symbols and User Controls in Movicon.NExT are WPF/XAML type. Therefore, the project screens created with XAML vector graphics are converted to SVG in HTML5 web pages when the project is deployed directly from the Movicon.NExT editor.

### Processing on Client side (Browser)

Originally the IIS Web Server was responsible for processing graphical pages displayed by connected Client Browsers within the WPF/HTML5 Web Client architecture. This meant that the size of IIS Web Server had to be quite substantial in order to support more than one Client Browser connection at the same time.

This has all changed with the new SVG/HTML5 Web Client architecture. Now, the integrated Web Server only makes HTML5 pages available to Browsers and uses the I/O Data Server to manage communications. Therefore, graphics are now processed on the Client (Browser) side giving the integrated Web Server leeway to support a higher number of Client connections simultaneously.

### Support to GeoScada and Augmented Reality functions

These two features are now available with the new WebHMI to allow use of Geographical Maps on par with those of the Client Desktop along with the use of Augmented Reality in combination with the Speech Recognition used in objects.

Thanks to these new additions, the WebHMI solution is now ideal to use as an alternative to the Client Desktop or Web Client (WPF/HTML5) and as a stand-alone solution on Windows and Linux operating system based HMI user panels.

### Supported Browsers

Movicon WebHMI has been test run on different platforms:

- **Windows:** Chrome, Edge.
- **iOS:** Safari, Chrome;
- **Android:** Chrome.

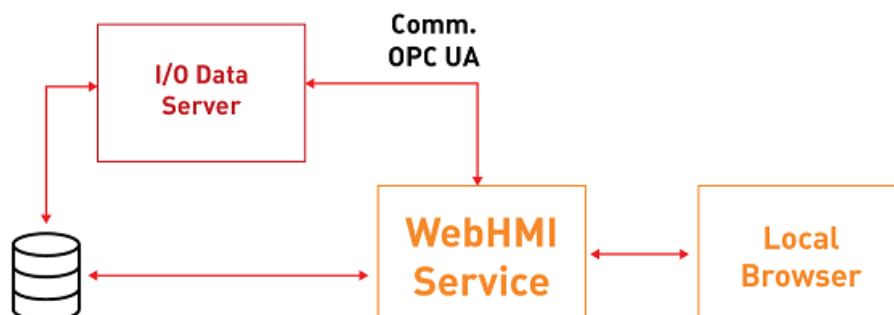
Support and the proper functioning of Movicon WebHMI cannot be guaranteed for all other browsers, such as Mozilla Firefox for example.



## 16.2. WebHMI Runtime Windows

WebHMI is the ideal solution for creating stand-alone applications in HMI with Windows or Linux operating systems.

The HMI can be managed by remote by predisposing the Deploy Server on the device and using the Deploy Project window and HMI Control directly from the Movicon.NExT development environment.



### Setting up HMI with Window OS

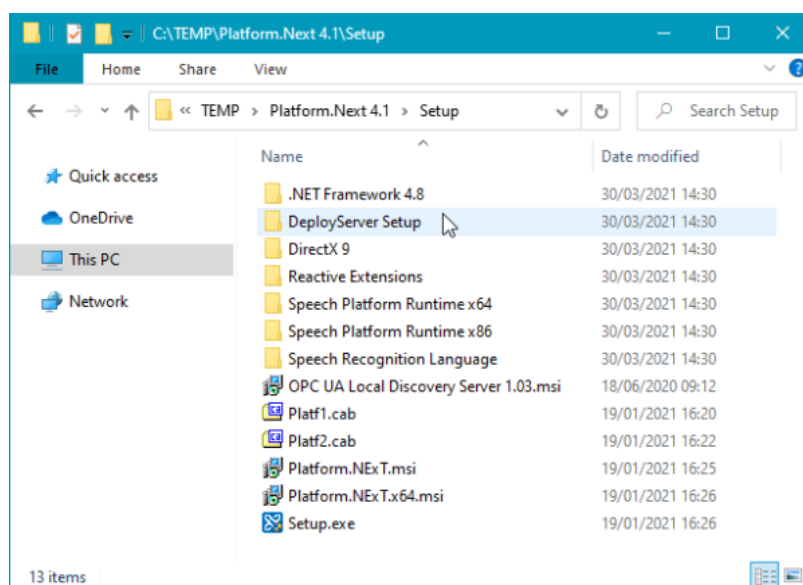
WebHMI allows you to create stand-alone applications in HMI with Windows operating systems natively compatible with .NET Core.

Components such as the .NET Core and Deploy Server, which consent I/O Server and WebHMI execution, are installed with the Setup.

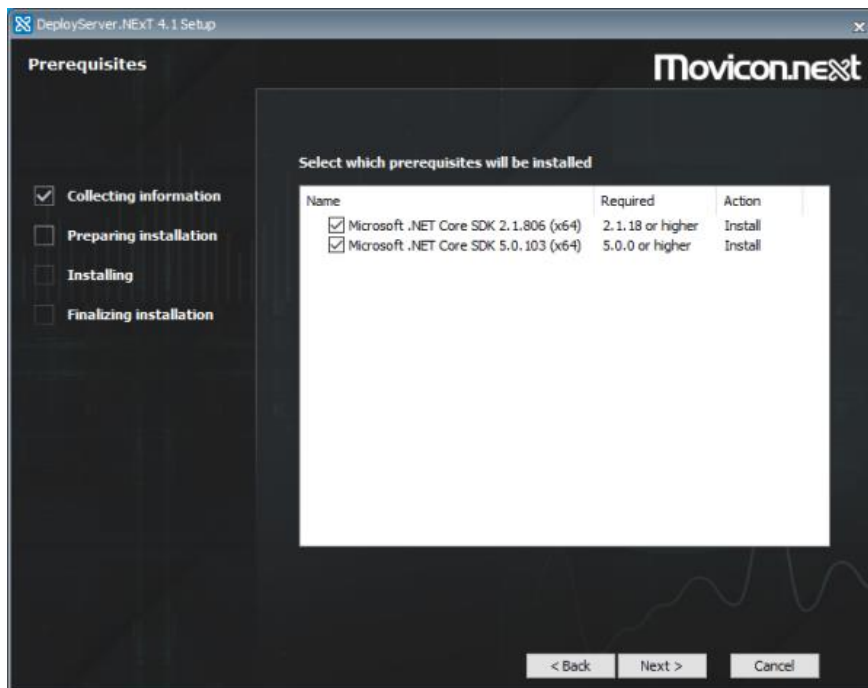
#### Deploy Server Installation

The Deploy Server allows you to download projects and manage devices by remote. Its installation also downloads the .NET Core versions needed to run it.

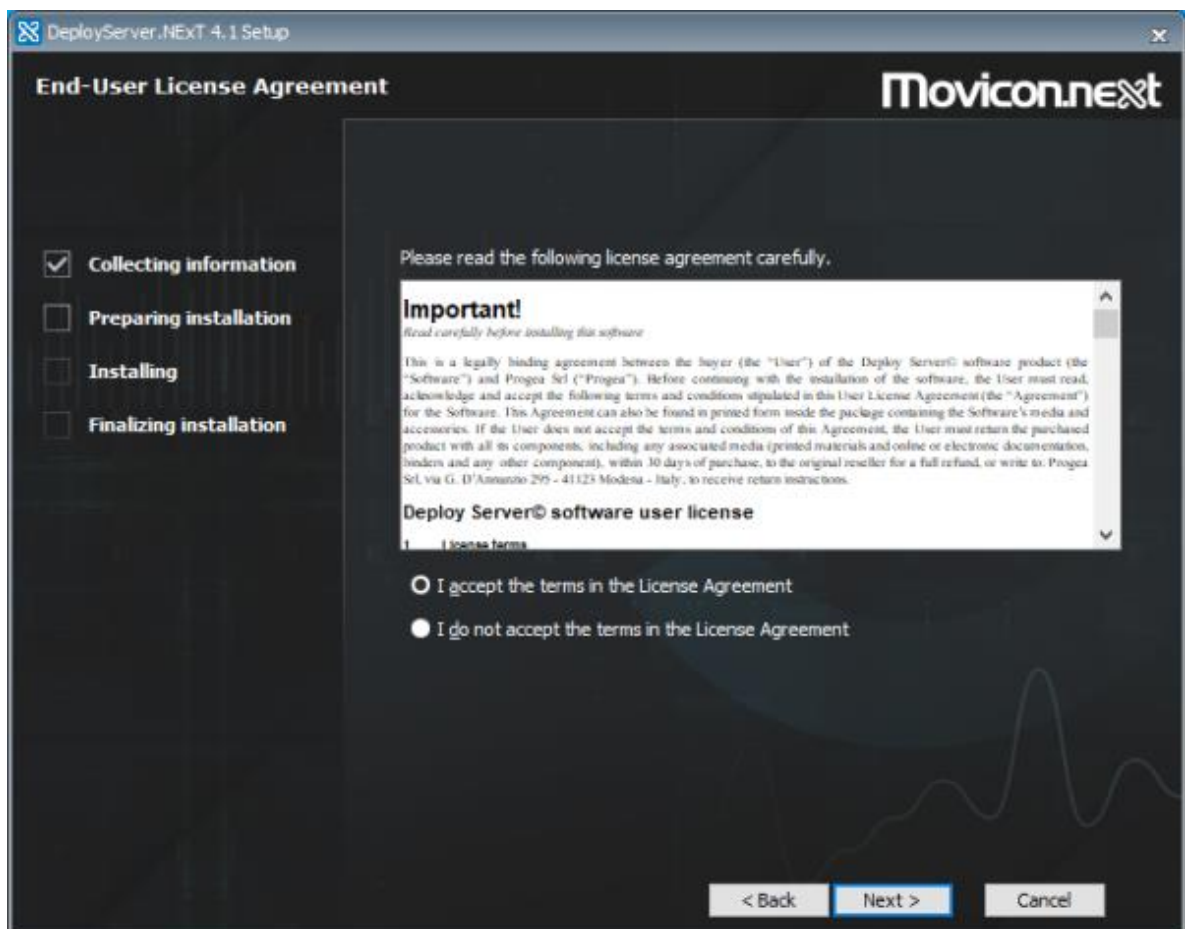
The Deploy Server setup is available from the "DeployServer Setup" folder contained in the Movicon.NExT setup:



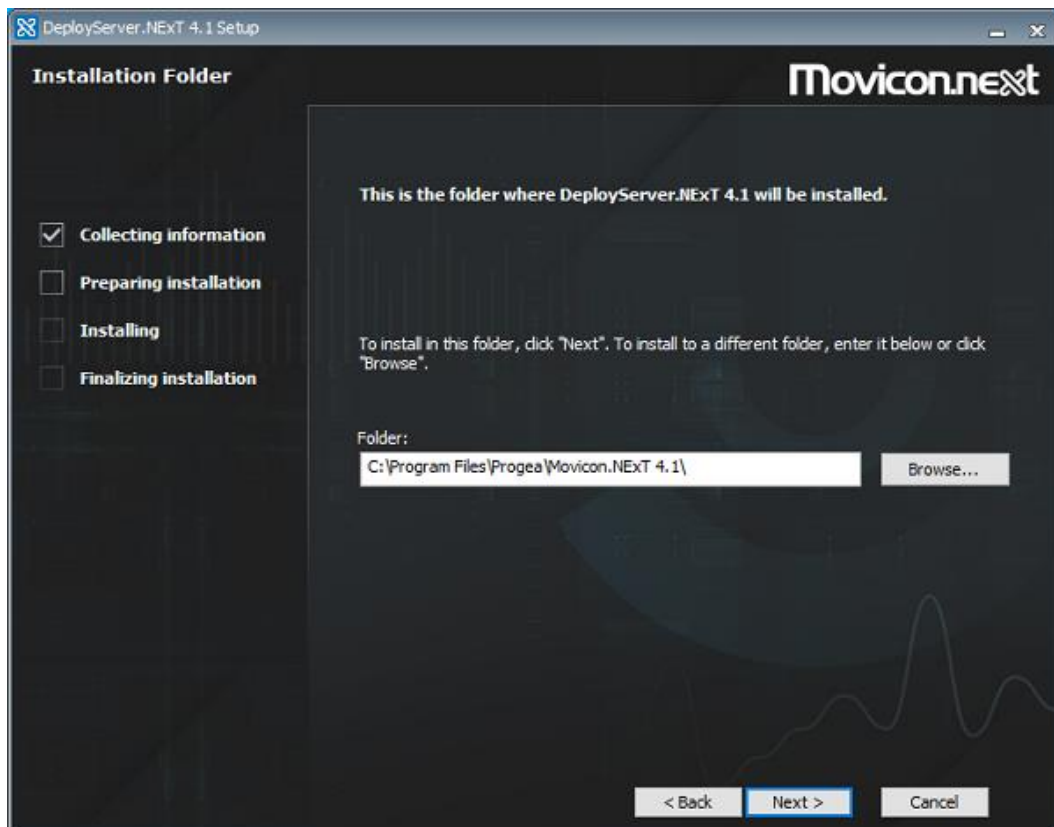
Once the Setup has been launched, a check will be run to see whether the .NET Core 2.1 and 5.0 packets, which are needed for running the Deploy Server and other components at Runtime, have been installed. If they have not been installed, the Setup will download them from the Microsoft website and install them.



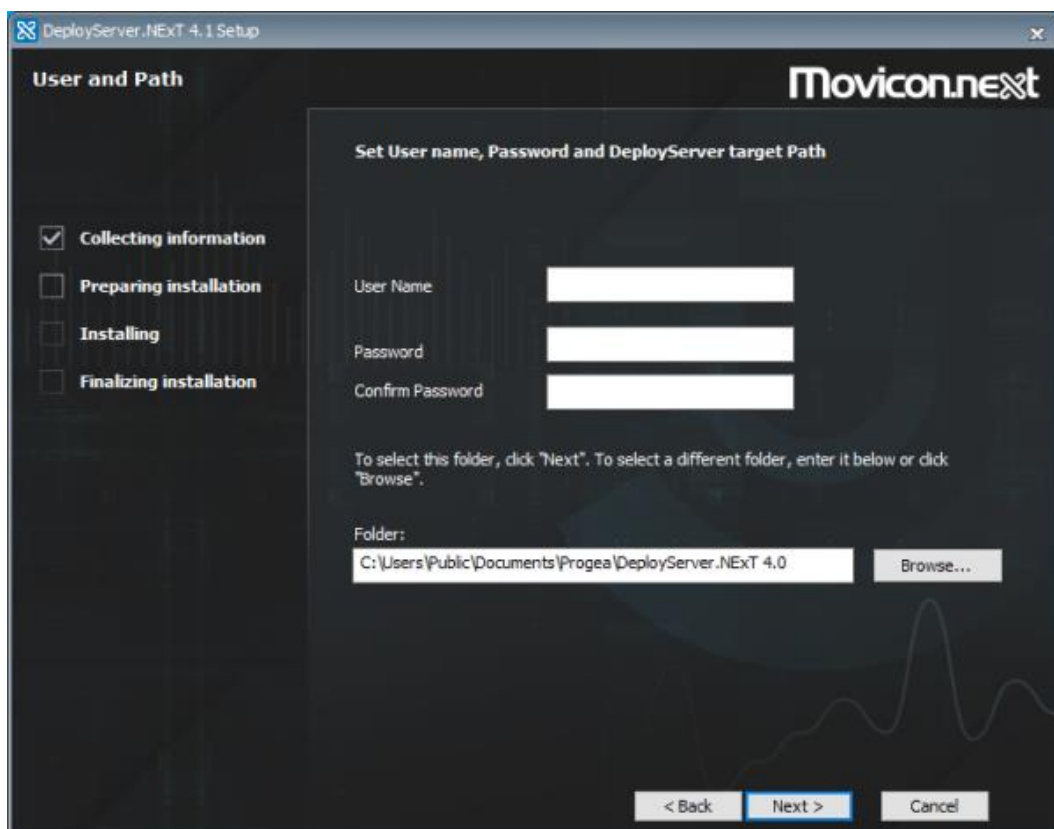
Once the prerequisites have been met, you can go ahead with installing the Deploy Server by accepting the terms of the user license agreement:



You will then be asked to enter the path in which to install the Deploy Server.



Following this, you will then be able to set the user and password to be used for connecting to the Deploy Server by means of the Deploy window, and the path in which the runtime components, which include the I/O Data Server and WebHMI Service and other components, will be installed on the device.



This last setup phase that asks for credentials and target folder can be avoided by launching the Deploy Server's setup from command line as follows:

```
DeployServerNext.exe DEPLOYUSERNAME={username}  
DEPLOYPASSWORD={password} DEPLOYPATH={absolute_path}  
For example:  
DeployServerNext.exe DEPLOYUSERNAME=admin@company.com  
DEPLOYPASSWORD=Password123$ DEPLOYPATH=C:\WebHMI
```

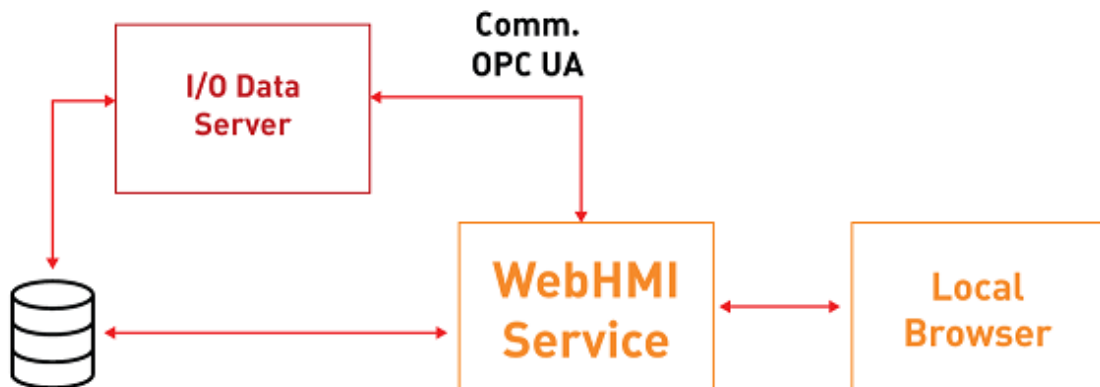
### Deploying projects on HMI

For further information on deploying projects on remote devices, please refer to "Deploying projects in HMI"

## 16.3. WebHMI Runtime Linux (x64)

WebHMI is the ideal solution for creating stand-alone applications in HMI with Windows or Linux operating systems.

The HMI can be managed by remote by predisposing the Deploy Server on the device and using the Deploy Project window and HMI Control directly from the Movicon.NExT development environment.



### Preparing HMI with Linux (x64) operating systems

WebHMI allows stand-alone applications to be created on HMI systems with Linux (64) operating systems compatible with .NET Core such as Ubuntu and Debian distributions.



The used Linux distribution must be compatible with .NET Core in order for the WebHMI to function.

Some components that consent the I/O Data Server and WebHMI to run within the Linux environment need to be installed manually as indicated below.

#### .NET Core Installation

1. Download and install the .NET Core SDK 2.1 on the HMI  
Go to this page: <https://dotnet.microsoft.com/download/dotnet-core/2.1> and choose which packets to install: Binaries or Installers.  
As an example, below are the commands to use in an Ubuntu 20.04 distribution to install .NET Core SDK 2.1:

```
wget https://packages.microsoft.com/config/ubuntu/20.04/packages-microsoft-prod.deb -O
packages-microsoft-prod.deb
sudo dpkg -i packages-microsoft-prod.deb
```

and subsequently:

```
sudo apt-get update; \
sudo apt-get install -y apt-transport-https && \
sudo apt-get update && \
sudo apt-get install -y dotnet-sdk-2.1
```

2. Download and install the .NET Core SDK 5.0 on the HMI  
Go to this page: <https://dotnet.microsoft.com/download/dotnet-core/5.0> and choose which packets to install: Binaries or Installers.  
As an example, below are the commands to use in an Ubuntu 20.04 distribution to install .NET Core SDK 5.0:

```
sudo apt-get update; \
sudo apt-get install -y apt-transport-https && \
sudo apt-get update && \
sudo apt-get install -y dotnet-sdk-5.0
```

### Database Management System Installation

4. The developer is free to choose which Database Management System (DBMS) they wish to install. However, the Microsoft SQL Server can be installed following the instructions at: <https://docs.microsoft.com/it-it/sql/linux/quickstart-install-connect-ubuntu?view=sql-server-linux-ver15>

### Deploy Server Installation

The Deploy Server allows you to download the project and manage the device by remote. Follow these steps below to install it:

5. Copy the contents of the "DeployServer-Files" folder from the Movicon.NExT installation folder (e.g., path "C:\Program Files\Progea\Movicon.NExT 4.0\DeployServer-Files") to the HMI.
6. Modify the "appsettings.json" file from the "DeployServer-Files" folder specifying:
  - username and password: which are to be then used in the deploy project window.

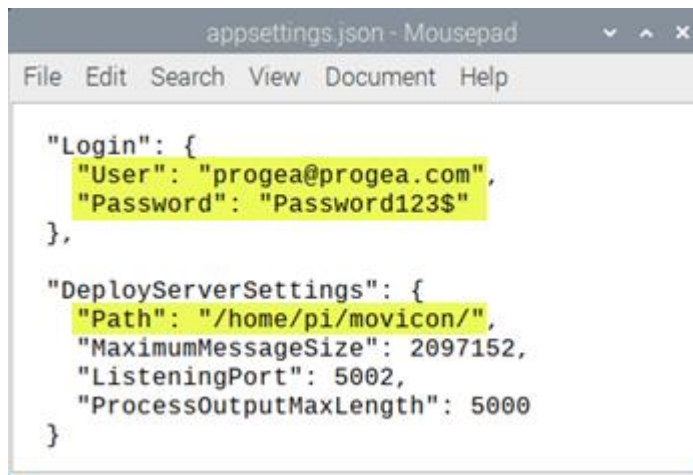


The username should be an existing or non existing email address, such as `progea@progea.com`



The password must be the minimum of 7 characters in length and must include a special character, an uppercase character and a number.

- path: the path in which the Deploy Server will save the deployed components (I/O Data Server, WebServer, Project)



```
appsettings.json - Mousepad
File Edit Search View Document Help

{
  "Login": {
    "User": "progea@progea.com",
    "Password": "Password123$"
  },
  "DeployServerSettings": {
    "Path": "/home/pi/movicon/",
    "MaximumMessageSize": 2097152,
    "ListeningPort": 5002,
    "ProcessOutputMaxLength": 5000
  }
}
```

7. Change the rights to the copied folder by giving full rights to the user who is using it (this step is optional and can be omitted by using the sudo command before all the following terminal commands).
8. Open a terminal window in the copied folder and run the "sudo dotnet dev-certs https" command to generate the certificate from the Deploy Server.
9. The Deploy Server can be started up manually with the "sudo dotnet DeployServer.dll" command or it can be configured to launch at the HMI startup as service.
  - a. When using the "sudo nano /etc/systemd/system/DeployServer.service" command it will create a .service file in the "/etc/systemd/system/" folder and it will open it directly with the text editor from the nano terminal.
  - b. Insert the following contents in the file by paying attention to the WorkingDirectory and ExecStart fields concerning the path where the Deploy Server was copied:

```
[Unit]
Description=Movicon NExT DeployServer

[Service]
WorkingDirectory=/home/pi/Documents/DeployServer-Files
ExecStart=dotnet /home/pi/Documents/DeployServer-Files/DeployServer.dll
Restart=always
RestartSec=2
SyslogIdentifier=DeployServer
User=root
Environment=ASPNETCORE_ENVIRONMENT=Production

[Install]
WantedBy=multi-user.target
```

- c. Use the "sudo systemctl daemon-reload" command (loads any service modifications)
- d. Use the "sudo systemctl enable DeployServer.service" command (enables the service just created)
- e. Use the "sudo systemctl start DeployServer.service" command (starts up service)
- f. Now if you use the "sudo systemctl status DeployServer.service" command, it will be active and running (checks the service status)



In order to install the license correctly, the Deploy Server must be started up with administration rights.



The Deploy Server listens on the 5002 (TCP) port. Therefore, you will need to check that the firewall permits communications on this port.

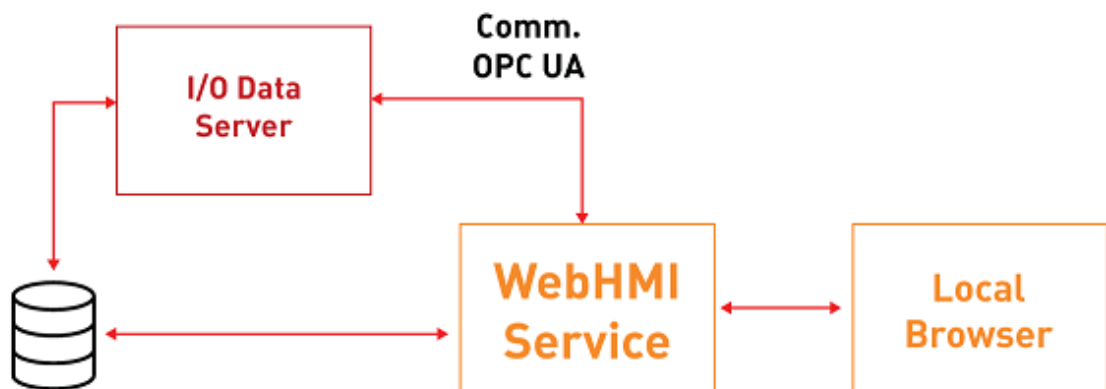
### Deploying Projects in HMI

For further information on deploying projects in remote devices, please refer to "Deploying Projects in HMI"

## 16.4. WebHMI Runtime Linux (ARM)

WebHMI is the ideal solution for creating stand-alone applications in HMI with Windows or Linux operating systems.

The HMI can be managed by remote by predisposing the Deploy Server on the device and using the Deploy Project window and HMI Control directly from the Movicon.NExT development environment.



### Preparing HMI with the Linux operating system (ARM)

WebHMI allows you to create stand-alone applications in HMI with the Linux operating system (ARM) that is compatible with .NET Core, such as for example Raspberry devices with Raspbian distribution.



The compatibility with .NET Core by the used Linux distribution is necessary for WebHMI to function.

Some components have been in the Linux environment allow I/O Data Server and WebHMI execution by installing them manually as indicated below.

#### .NET Core installation

1. Download and install the .NET Core SDK 2.1 on the HMI  
To download the Binaries for ARM, go to this page: <https://dotnet.microsoft.com/download/dotnet-core/2.1>  
For Raspbian, download: <https://download.visualstudio.microsoft.com/download/pr/47cf1500-e5de-4fb6-8506-2818beda75bc/288834aa886d9374acca223591413435/dotnet-sdk-2.1.806-linux-arm.tar.gz>

2. Download and install the .NET Core SDK 5.0 on the HMI.  
To download the Binaries for ARM, go to this page:  
<https://dotnet.microsoft.com/download/dotnet-core/5.0>  
For Raspbian, download: <https://download.visualstudio.microsoft.com/download/pr/5c2e5668-d7f9-4705-acb0-04ceeda6dadf/4eca3d1ffd92cb2b5f9152155a5529b4/dotnet-sdk-5.0.103-linux-arm64.tar.gz>
3. Download the "InstallDotNet.sh" script and copy it in the same folder where the two previously downloaded tar.gz archives have been saved.
4. Check that the pi user is present on the device, otherwise change the paths present in the "InstallDotNet.sh" file.
5. Copy the downloaded files (tar.gz) and the script (InstallDotNet.sh) in a local device folder as it will not be possible to run them from an external device (e.g. USB dongle).
6. Open the terminal on the device and go to the folder in which the "InstallDotNet.sh" file resides.
7. Use the "chmod +x InstallDotNet.sh" command.
8. Use the "./InstallDotNet.sh" command.
9. Check that both versions are installed and acknowledged correctly with the "dotnet --info" command.

### Database Management System Installation

10. It is up to the developer to choose which Database Management System (DBMS) to install. However, a dedicated MySQL called MariaDB is available for Raspbian/ARM for which the installation commands are.  

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install mariadb-server
sudo mysql_secure_installation
```

For further information please go to: <https://pimylifeup.com/raspberry-pi-mysql/>

### Deploy Server Installation

10. Copy the contents from the "DeployServer-Files" folder, present in the Movion.NExT installation folder (e.g. at path: "C:\Program Files\Progea\Movicon.NExT 4.0\DeployServer-Files") to the HMI.
11. Modify the "DeployServer-Files" folder's "appsettings.json" file specifying:
  - username and password: which must then be used in the project deploy window.



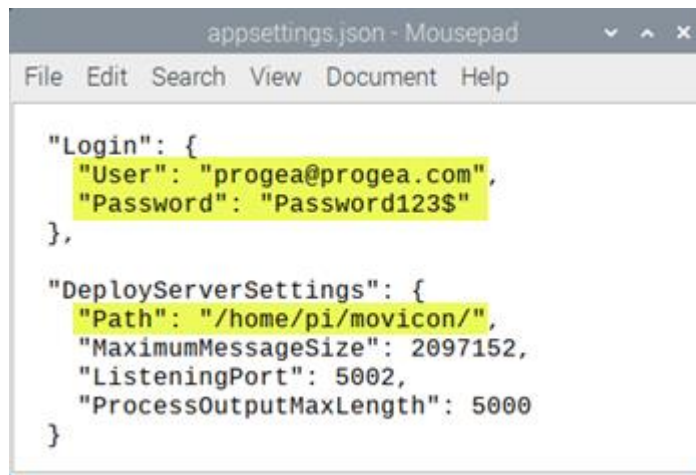
The username should be an existing or non existing email address, such as progea@progea.com



The password must be the minimum of 7 characters in length and must include a special character, an uppercase character and a number.

5. path: the path in which the Deploy Server will save the transferred components (I/O Data Server, WebServer and Project)





```
appsettings.json - Mousepad
File Edit Search View Document Help

{
  "Login": {
    "User": "progea@progea.com",
    "Password": "Password123$"
  },
  "DeployServerSettings": {
    "Path": "/home/pi/movicon/",
    "MaximumMessageSize": 2097152,
    "ListeningPort": 5002,
    "ProcessOutputMaxLength": 5000
  }
}
```

12. Change the rights to the copied folder by give full rights to the user who is using it (this step is optional, it may be omitted by using the sudo command before all the following terminal commands).
13. Open the terminal window folder you have just copied and run the "sudo dotnet dev-certs https" command to generate the certificate from the Deploy Server.
14. To startup the Deploy Server, you can use the "sudo dotnet DeployServer.dll" command to start it up manually, or configure it in a way that it is launched at the HMI startup as service.
  - a. When using the "sudo nano /etc/systemd/system/DeployServer.service" command it will create a .service file in the "/etc/systemd/system/" folder and it will open it directly with the text editor from the nano terminal.
  - b. Insert the following contents in the file by paying attention to the WorkingDirectory and ExecStart fields concerning the path where the Deploy Server was copied:

```
[Unit]
Description=Movicon NExT DeployServer

[Service]
WorkingDirectory=/home/pi/Documents/DeployServer-Files
ExecStart=dotnet /home/pi/Documents/DeployServer-Files/DeployServer.dll
Restart=always
RestartSec=2
SyslogIdentifier=DeployServer
User=root
Environment=ASPNETCORE_ENVIRONMENT=Production

[Install]
WantedBy=multi-user.target
```

- c. Use the "sudo systemctl daemon-reload" command (loads any service modifications)
- d. Use the "sudo systemctl enable DeployServer.service" command (enables the service just created)
- e. Use the "sudo systemctl start DeployServer.service" command (starts up service)
- f. Now if you use the "sudo systemctl status DeployServer.service" command, it will be active and running (checks the service status)



In order to install the license correctly, the Deploy Server must be started up with administration rights.



The Deploy Server listens on the 5002 (TCP) port. Therefore, you will need to check that the firewall permits communications on this port.

### Deploying Projects in HMI

For further information for transferring projects to remote devices, please refer to "Deploying Projects in HMI"

## 16.5. Deploying projects in HMI

This topic shows you how to transfer the project to a remote device after having configured the Deploy Server as described in the previous topics.

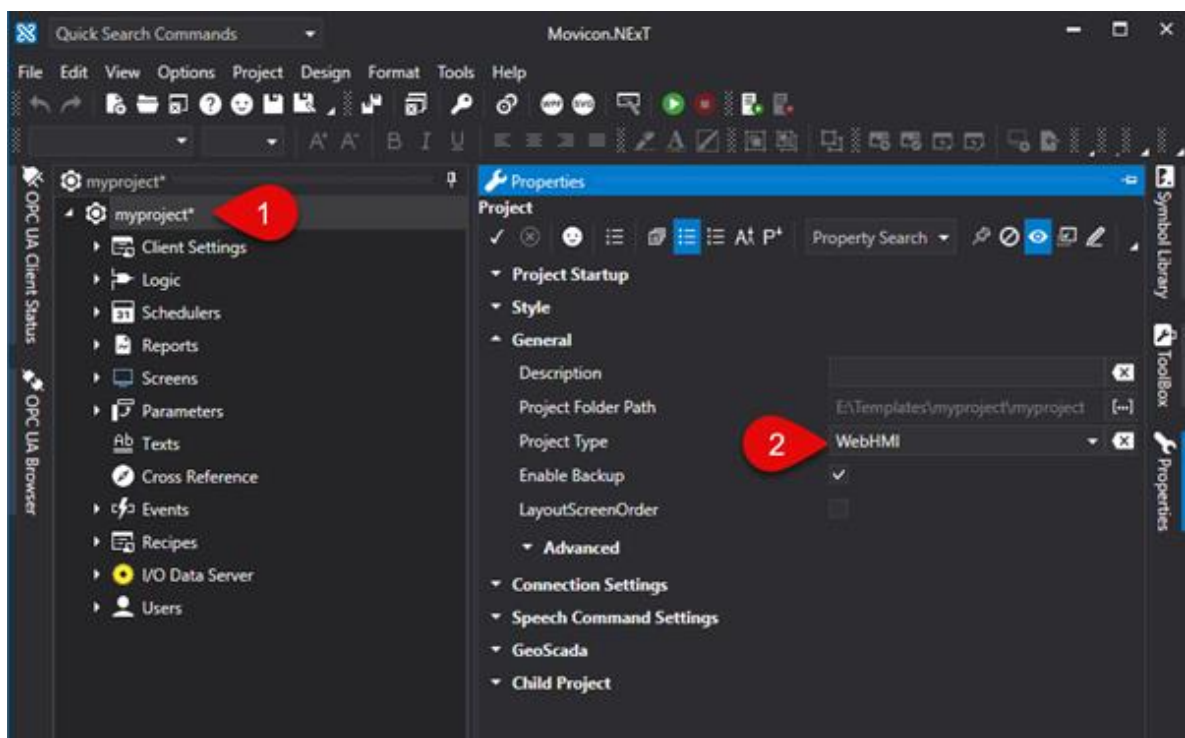
The steps to deploy the project correctly are:

- check the project type and toolbox objects used on screens.
- convert project to SVG
- open Deploy window
- connect to the device's Deploy Server and transfer components

### Property: Project Type

The "Project Type" property is one of the project properties that is used to set the project as a "WebHMI" type.

When setting the project as "WebHMI", the Project Explorer, Toolbox and Animation Explorer and Command windows will only show the resources, animation objects and commands that are supported while visualizing and using Web Client WebHMI.



This property setting is particularly useful when:

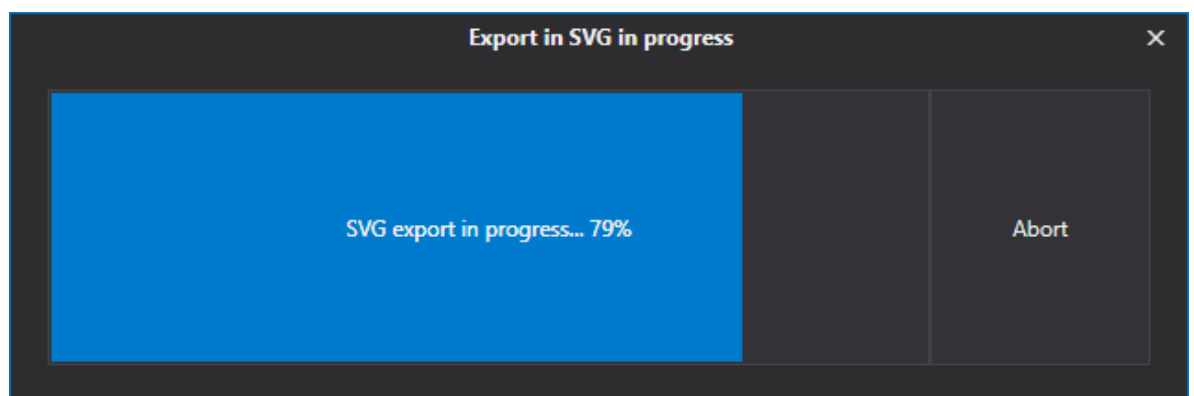
- planning to develop a project that includes only those functions supported by WebHMI,

- needing to verify the compatibility of a previously developed project with WebHMI visualization. Any objects and symbols used in the project and not supported by WebHMI will be highlighted as shown in the screenshot below.

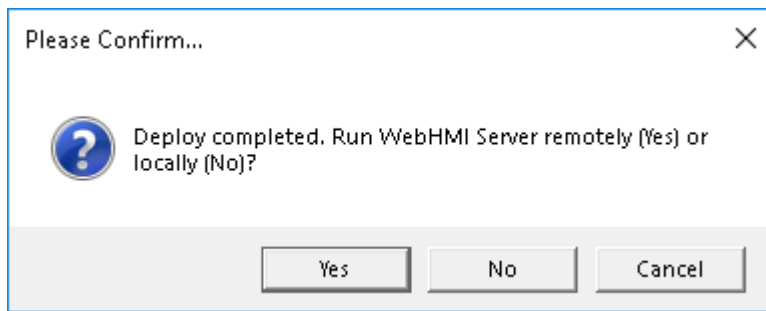


## Project Conversion

To deploy a project in remote HMI, simply select the command from the Options > Create WebClient (SVG/HTML5) menu or select Create WebClient (SVG/HTML5) from the project's toolbar that starts the project's export to SVG:



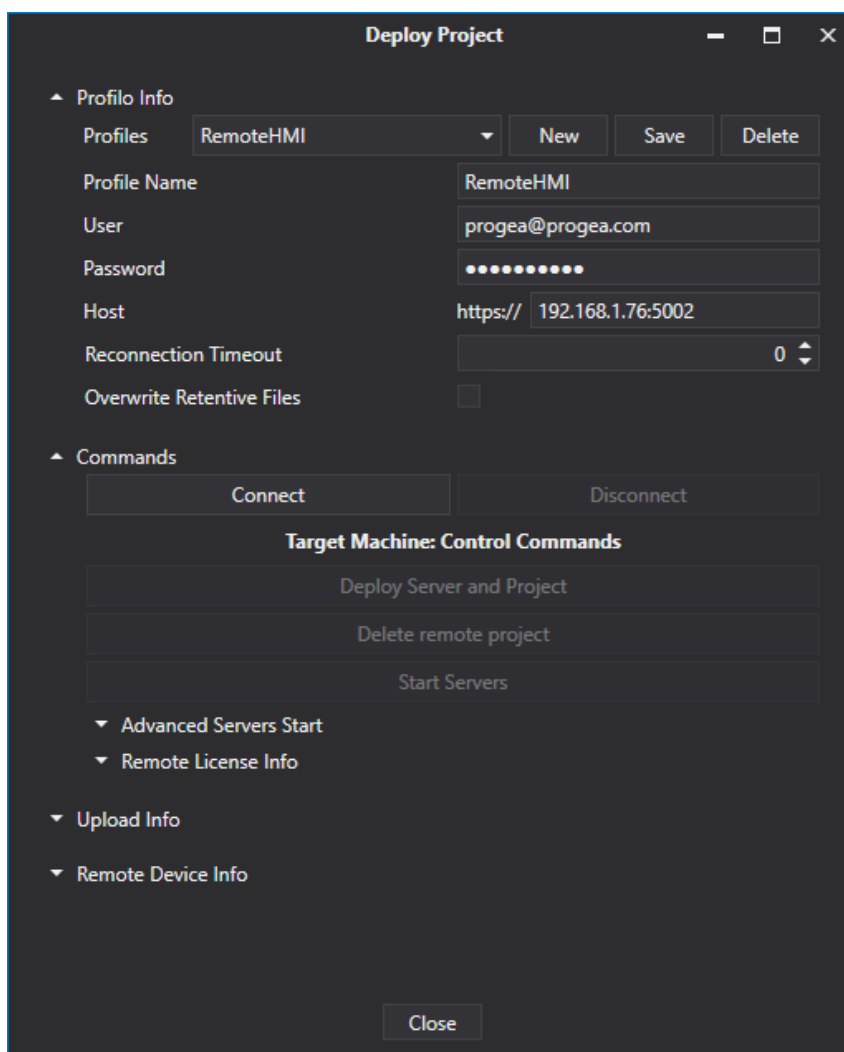
Once the export process has terminated, you will be prompted to confirm whether or not to deploy and run the Web Server and Project on remote HMI as shown below:



Selecting "Yes", a Deploy Project window will open as indicated in the next paragraph.

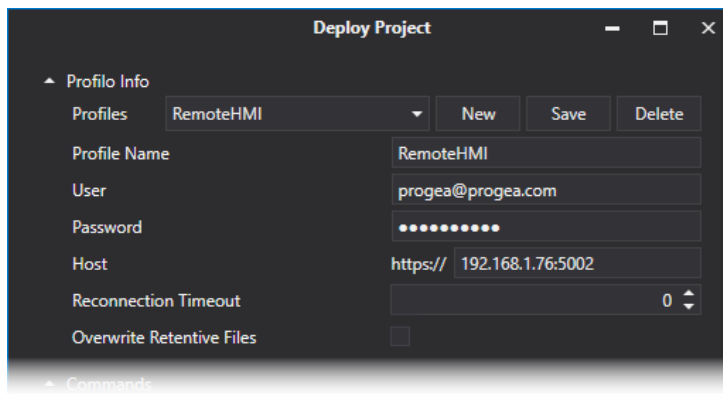
## The Deploy Project Window and Remote HMI Control

The Deploy Project window allows you to deploy the project on the HMI along with the I/O Data Server and Web Server which are needed to run the WebHMI locally on the HMI.



### Profile Info

This section allows you to define different connection profiles, such as for Deploy Servers of different devices for instance.

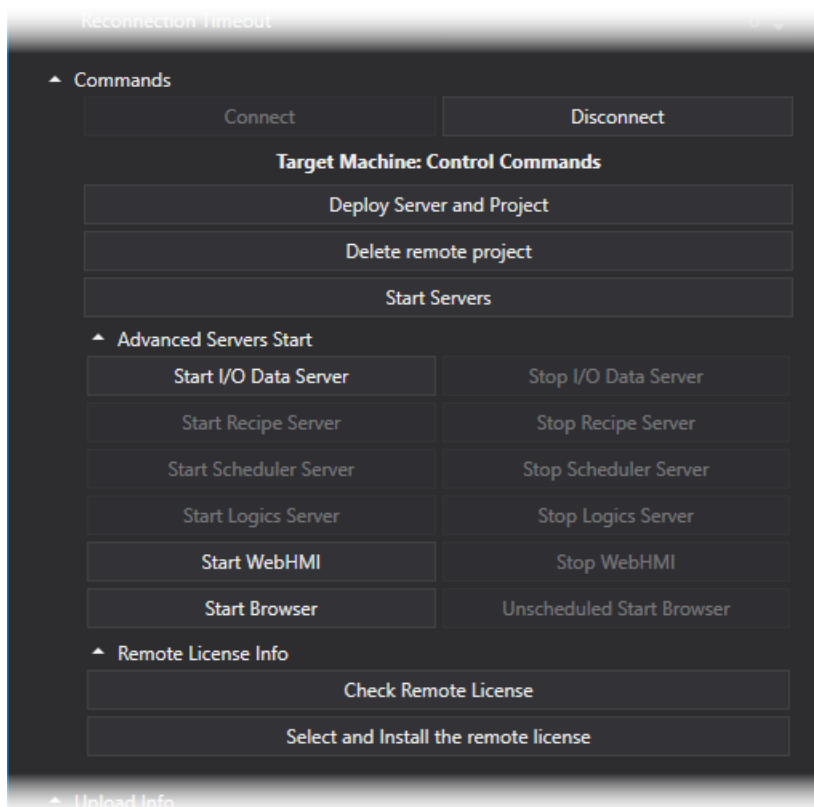


where:

- **Profile name:** the profile name which is an unique identifier used by the developer to distinguish the various profiles.
- **User:** user name with which to connect to the target HMI's Deploy Server.
- **Password:** password to use for connecting to the target HMI's Deploy Server.
- **Host:** HMI's IP address or host name, including the Deploy Server port (E.g: 192.168.0.74:5002)
- **Overwrite Retentive Files:** retentive data in the device will be overwritten with those in the project during the project deployment phase. The retentive files are located in the project folders:
  1. ...\\UFUAServer\\Alarms
  2. ...\\UFUAServer\\Tags
  3. ...\\UFUAServer\\EventLog
  4. ...\\UFUAServer\\Historian
  5. ...\\UFUAServer\\DataLogger

### Commands:

From this section it is possible to manage the connection to the Deploy Server, control the Runtime components and manage the license on the remote device.



were:

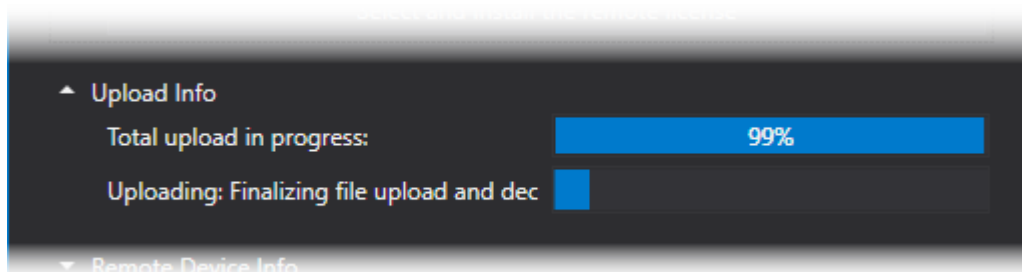
- **Connect/Disconnect:** This allows you to connect/disconnect from the HMI's Deploy Server. While connecting to the target machine, the client will check and verify whether there are any processes started up belonging to project other than the current one and, in if so, the user will be asked, by means of a popup window as shown below, whether they should be replaced. If user confirms this, all the pre-existing processes in the device of the different projects will be aborted. Otherwise, they will remain unchanged.
- **Deploy Project and Server:** deploys the Project, I/O Data Server and WebHMI's Web Server on the target HMI.
- **Cancel Remote Project:** this cancels the project on the target HMI.
- **Server Startup:** Starts all the services on the target HMI.
- **Advanced Server Startup:** allows the execution of each Runtime component to be started up or stopped.
- Remote License Info > **Check Remote License:** displays the remote HMI's active license options and Site Code
- Remote License Info > **Install Remote License:** this allows the software license to be installed on the remote HMI.



When first connecting to the Target HMI and after the Self Signed certificate has been accepted, you will be asked to download the I/O Data Server, the WebHMI's Web Server and the project. If they are already present on the HMI, you will be asked if you wish to update them.

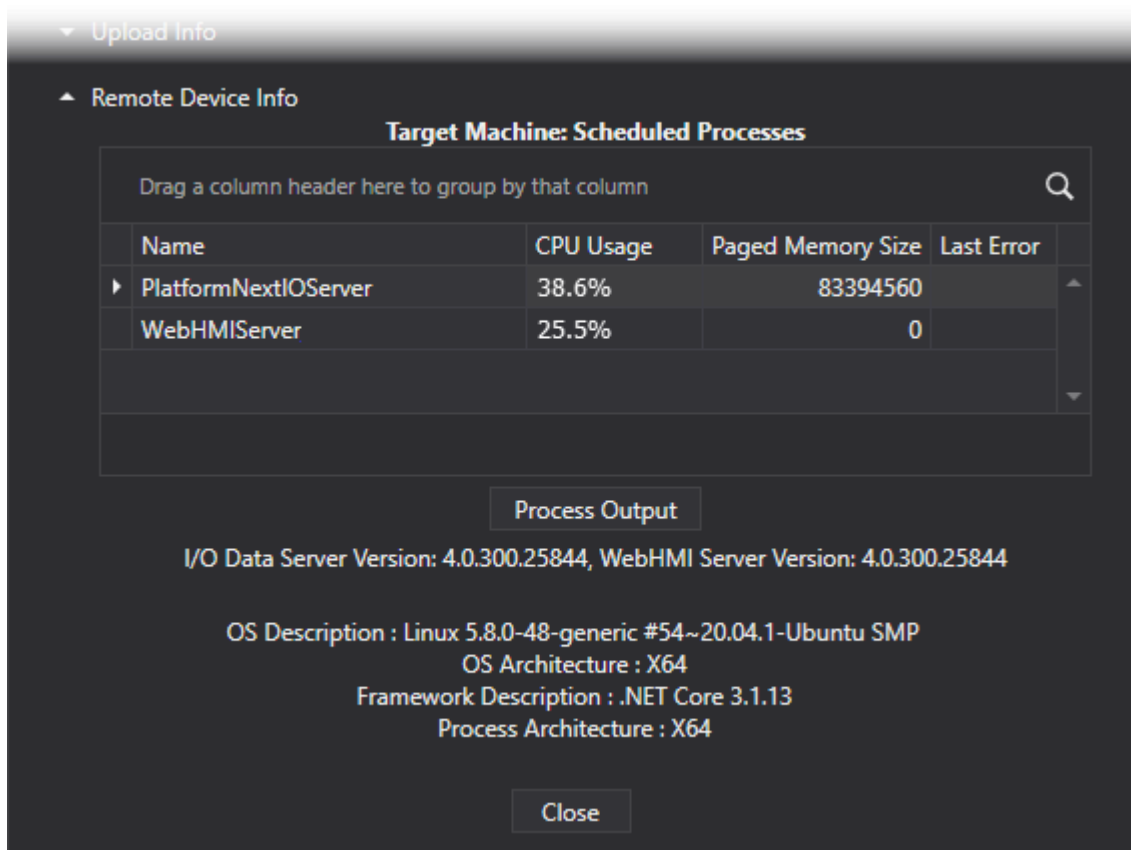
#### Upload Info:

This section reports the progress of the transfer in progress.



#### Remote Device Info:

This shows information on the remote device's processes and operating system.



where:

- **Scheduled Processes:** shows the process running on the remote HMI and CPU and RAM usage.
- **Process Output:** reports the actual Syslog file contents on the remote device.
- **OS Description:** shows the versions of the installed Runtime components and information on the HMI's operating system.

### Connecting with Browser

The project is deployed by the integrated Web Server on ports 5000 (http) and 5001 (https). The default configuration requires that the requested received on port 5000 be automatically redirected to port 5001. To change these settings please refer to chapter on "Integrated Web Server Settings" .

The following addresses can be used to access pages exposed by the Web Server:

http://<WebServer\_IP\_Address>:5000  
 https://<WebServer\_IP\_Address>:5001

Example: https://192.168.10.10:5001

It is also possible to open a different screen to use as the start page instead of the one defined as the Main Page in the project. To do this simply add the name of the screen to be opened to the url using the following syntax:

https://<WebServer\_IP\_Address>:5001/?page=<ScreenName>

where <ScreenName> is the name of the Screen without an extension. If the screen is inside a folder, the name of that folder should also be specified.

Example:

`https://192.168.10.10:5001/?page=Screen1`

`https://192.168.10.10:5001/?page=Folder1/Screen1`

Ultimately, a parameter file can also be passed while loading the specified screen. In order to do this, you must add the name of the parameter file to the url after the name of the screen:

`https://<WebServer_IP_Address>:5001/?page=<ScreenName>&parameter=<ParameterName>`

where <ParameterName> is the name of the Parameter file without an extension. If the parameter file is inside a folder, you should also specify the name of that folder.

Example:

`https://192.168.10.10:5001/?page=Screen1&Parameter1`

`https://192.168.10.10:5001/?page=Folder1/Screen1&Folder1/Parameter1`

In addition, the WebHMI page can be opened by performing user logon directly from the URL. To do this, simply add the username and password to the URL using the following syntax:

`https://<WebServer_IP_Address>:5001/?user=<UserName>&psw=<UserPassword>`

dove <UserName> is the name of the user who will logon and the <UserPassword> is the user's password.

Example:

`https://192.168.10.10:5001/?user=User1&psw=12345`

User logon can also be performed using a specific screen as follows:

`https://192.168.10.10:5001/?page=Screen1&parameter=Parmeter1&user=User1&psw=12345`

## 16.6. Integrated Web Server Settings

The WebHMI's integrated Web Server settings can be changed by editing the "appsettings.json" file which is found in "C:\Program Files\Progea\Movicon.NExT 4.0\WebHMIServer\"

The customizations available are listed below in the following paragraphs.

### Changing the listening port (default 5000 e 5001)

The default listening port can be changed by modifying the values of the "HttpListeningPort" and "HttpsListeningPort" parameters.

### Deactivation of the http or https protocol

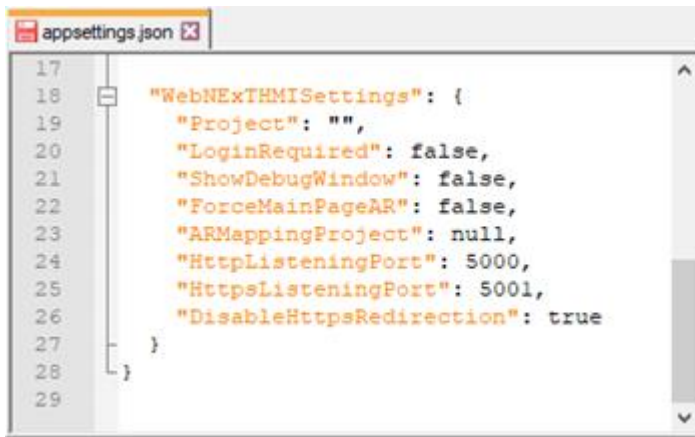
When necessary, it is possible to select and set the respective parameter to zero of either the http or https protocol to deactivate it.

For example: "HttpsListeningPort": 0

### Deactivating the redirection from port 5000 to port 5001 (active for default)

The redirection from port 5000 to port 5001 can be deactivated by adding the "DisableHttpsRedirection" parameter to the "WebNExTHMISettings" group and setting it to True come as shown in the image below.





### Using a customized SSL certificate

A customized certificate can be used in substitution of the automatically generated self-signed certificate by adding the following rows to the "appsettings.json" file:

```
"Kestrel": {
  "Endpoints": {
    "HTTP": { "Url": "http://*:5000" },
    "HTTPS": {
      "Url": "https://*:5001",
      "Certificate": {
        "Path": "certificate_file.pfx",
        "Password": "certificate_password"
      }
    }
  }
}
```

where "certificate\_file.pfx" is the name of the certificate including the absolute path.

It is possible to use a relative path with respect to the "WebNExTHMI.exe" executable, as in the case of the example.

## 16.7. Security Certificate Management

When the user management is activated or the Scheduler Viewer is used in a WebHMI project (or any resource deployed using the WebHMI service), you will need to copy the security certificated in use in the trusted certificates folder for the WebHMI service.

The certificates to copy are:

- "Platform.NExT.v2 IOServer [xxx].der" = I/O Data Server certificate
- "Platform.NExT.v2 Scheduler [xxx].der" = Scheduler Server certificate
- "Platform.NExT.v2 Web Server [xxx].der" = WebHMI Server certificate

The path where these certificates are found varies according to the operating system where the services being run.



If the certificates are already present in the WebHMI service's trusted certificates folder, you will not need to do anything.

## Window Operating System

The service certificates in Windows are found in:

`%ProgramData%\OPC Foundation\CertificateStores\MachineDefault\certs`

and the respective ".der" files relating to the services used should be copied in the WebHMI service's trusted certificates folder which is different depending on the user who runs it.

If the LocalSystem is the user that launches the WebHMI Service, the folder will be:

`%windir%\System32\config\systemprofile\AppData\Local\OPC Foundation\pki\trusted\certs`

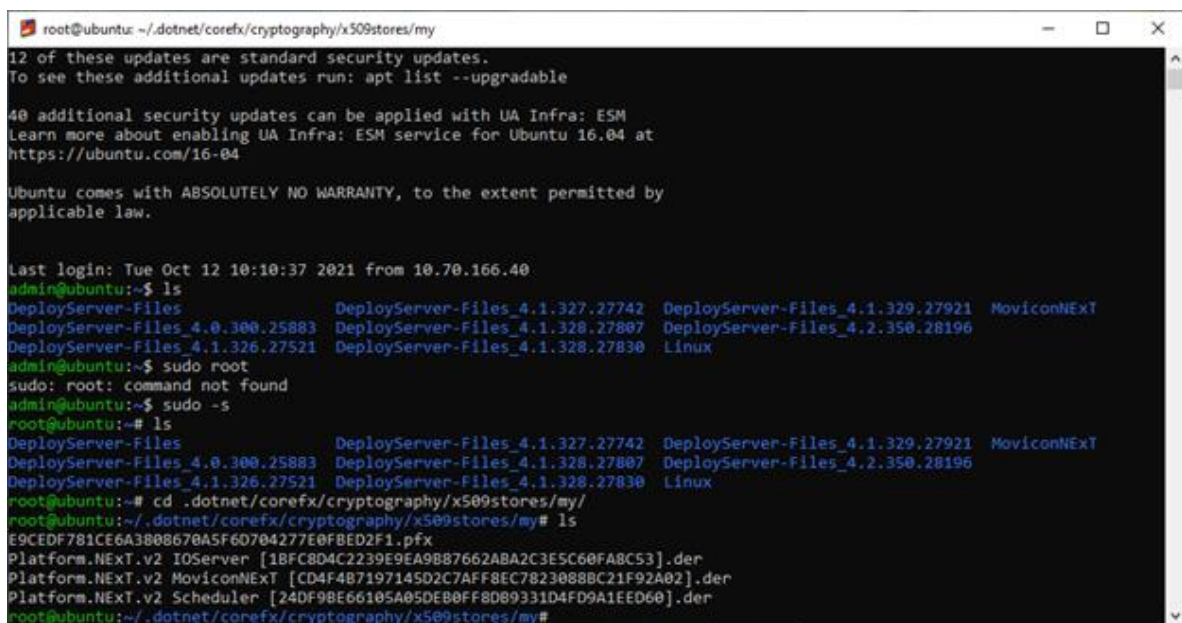
If the current Windows user is the one that launched the WebHMI service, the folder will be:

`%localappdata%\OPC Foundation\pki\trusted\certs`

## Linux Operating System

The certificates for the Movicon.NExT services in Linux are found in:

`root/.dotnet/corefx/cryptography/x509store/my/`



```
root@ubuntu: ~/.dotnet/corefx/cryptography/x509stores/my
12 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

40 additional security updates can be applied with UA Infra: ESM
Learn more about enabling UA Infra: ESM service for Ubuntu 16.04 at
https://ubuntu.com/16-04

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Tue Oct 12 10:10:37 2021 from 10.70.166.40
admin@ubuntu:~$ ls
DeployServer-Files      DeployServer-Files_4.1.327.27742  DeployServer-Files_4.1.329.27921  MoviconNExT
DeployServer-Files_4.0.300.25883  DeployServer-Files_4.1.328.27807  DeployServer-Files_4.2.350.28196
DeployServer-Files_4.1.326.27521  DeployServer-Files_4.1.328.27830  Linux
admin@ubuntu:~$ sudo root
sudo: root: command not found
admin@ubuntu:~$ sudo -s
root@ubuntu:~# ls
DeployServer-Files      DeployServer-Files_4.1.327.27742  DeployServer-Files_4.1.329.27921  MoviconNExT
DeployServer-Files_4.0.300.25883  DeployServer-Files_4.1.328.27807  DeployServer-Files_4.2.350.28196
DeployServer-Files_4.1.326.27521  DeployServer-Files_4.1.328.27830  Linux
root@ubuntu:~# cd .dotnet/corefx/cryptography/x509stores/my/
root@ubuntu:~/.dotnet/corefx/cryptography/x509stores/my# ls
E9CEDF781CE6A3B08678A5F6D704277E0FBED2F1.pfx
Platform.NExT.v2 IOserver [18FC8D4C2239E9EA9B87662ABA2C3E5C60FA8C53].der
Platform.NExT.v2 MoviconNExT [CD4F4B7197145D2C7AFF8EC7823088BC21F92A02].der
Platform.NExT.v2 Scheduler [24DF98E66105A05DE0FF8D089331D4FD9A1EE060].der
root@ubuntu:~/.dotnet/corefx/cryptography/x509stores/my#
```

and the respective ".der" files relating to the services in use are to be copied in the WebHMI service's trusted certificates folder found in:

`root/.local/share/OPC Foundation/pki/trusted/certs/`

```

root@VnUbuntuSupport:~# find / OPC | grep OPC
find: '/run/user/1000/gvfs': Permessonegato
/root/.local/share/OPC Foundation
/root/.local/share/OPC Foundation/pki
/root/.local/share/OPC Foundation/pki/rejected
/root/.local/share/OPC Foundation/pki/rejected/certs
/root/.local/share/OPC Foundation/pki/rejected/certs/UaExpert@GIORDANOT [08164BEF64625CC16F17A449CEFE0D21A14343E].der
/root/.local/share/OPC Foundation/pki/trusted
/root/.local/share/OPC Foundation/pki/trusted/certs
/root/.local/share/OPC Foundation/pki/trusted/certs/Platform.NExT.v2 IOserver [C62D00AC16454ED29D398319D8DB47182769418C].der
/root/.local/share/OPC Foundation/pki/trusted/certs/Uaexpert.der
/root/.local/share/OPC Foundation/pki/trusted/certs/Platform.NExT.v2 MoviconNExT [A5AAB7FCF3C66BAAC565921398F022F4EE759B1D].der
/home/progea/Movicon/IODataServer/OPCUAViewModel.dll
/home/progea/Movicon/IODataServer/OPCUAUtilities.dll
/home/progea/Movicon/IODataServer/OPCUAViewModel.dll.config
/home/progea/Movicon/IODataServer/OPCUAClient.Config.xml
/home/progea/Movicon/WebHMIServer/OPCUAViewModel.dll
/home/progea/Movicon/WebHMIServer/OPCUAUtilities.dll
/home/progea/Movicon/WebHMIServer/OPCUAViewModel.dll.config
/home/progea/Movicon/WebHMIServer/OPCUAClient.Config.xml
/home/progea/Movicon/Projects/TestOPCCert
/home/progea/Movicon/Projects/TestOPCCert/Log
/home/progea/Movicon/Projects/TestOPCCert/Log/SysLog.log
/home/progea/Movicon/Projects/TestOPCCert/Log/SysLog.xml

```



This folder is available if access to the Linux machine was performed as "root" user.

In cases where the I/ODataServer running on Linux has to be reachable by an external OPC-UA client, the relative client certificate should be moved to this folder. This will also require activation of the license "OPC UA Server" option.

## 16.8. WebHMI Functions and Limitations

Below are listed the features supported by WebHMI. Any features not shown on this list are not supported.

### Supported Objects from the Toolbox

- Alarms: Alarm Banner, Alarm Window (without use of filters), Historian Log Viewer
- Progress Bars
- Buttons: standard and with styles
- DB Connectors: Combo-box, Grid Control
- Dialog Boxes: Check-box, Combo-Box, Spin button
- Displays
- Arc Bargraphs: all
- Gauge: circular e transparent 180°
- Meter
- Gauge Advanced
- Hot Regions
- Dynamic Images
- Lights: all
- Bargraphs: all
- Bargraph Advanced: all
- Draws: all
- Recipes: Recipe Viewer
- Schedulers: Scheduler Viewer
- Selectors: 2 position selectors, 3 position selectors (0-1-2 e 1-0-2)
- Sliders: horizontal, vertical, flat horizontal, flat vertical
- Switches: Push button switches, toggle switches
- Screen Tabs
- Texts: Animated Text, Text Box
- Trends: Chart, Trend Data Analysis, Real Time Trend, State Bar Chart, Sparkline Chart

- Viewers: Report Viewer, PDF Viewer
- Historian Data Viewers: Data Logger Viewer, Historian Viewer



The "viewer" objects of the Toolbox, such as "Alarm Window", "Event Log", "Grid", "Data Analysis Trend", etc., on the WebHMI side, can be presented with a reduced set of available commands compared to the Client Desktop version. In principle, the following may be missing:

- some commands of the object toolbar
- some command buttons
- filter functionality
- data export or print functionality
- pop-up windows (e.g. Palette window or Data Analysis Grid)

### Supported Project Resources

- Logic
- Schedulers
- Reports
- Screens
- Parameters
- Texts
- Cross Reference
- Events (excluding Events linked to System tags or local Client tags. They are supported only Events linked to I/O Data Server Tags)
- Recipes
- IO Data Server
- Users (excluding Domain users, and it is not possible to share / synchronize users between clients of different types)

### Supported Animations



Animations are only supported by the symbols from the Symbol Gallery and objects from the "Draws" group in the Toolbox.

- Background Color
- Visibility
- Blink
- Filling
- Height
- Width
- Scale
- Rotation
- Move X
- Move Y

### Supported Commands

- Screens (except for the Auto Close and Monitor ID options)
- Tag Value
- Open Pad
- Recipes (except for commands: Export/Import and Show Recipe)
- Users
- Change Language
- Reports

- Alarms
- Text to Speech
- Open Map

#### Other limitations

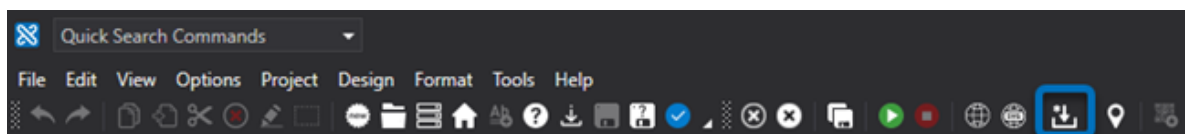
- The .NET Core framework support is needed in stand-alone applications (WebHMI Runtime)
- Only USB hardware or software licences are supported in stand-alone applications (WebHMI Runtime), and not NET licenses.
- vb.net scripts are not supported on both server and client sides in stand-alone applications (WebHMI Runtime), but logic is available.
- Only scripts on server side run by the Script service are supported when using WebClient only (SVG/HTML5).
- Projects cannot be saved on database.
- The I/O Data Server redundancy is not supported.,
- Data validation is not available for data recorded with the "Audit Trail" (CFR21),
- 3D graphics are not supported.
- Multitouch functions are not supported.
- "opc.tcp" and "http" are the only client-server communication transports available.
- Log files are saved as text files and not xml. files.
- "tiff" images are not supported.

## 16.9. How to activate a WebHMI Runtime License

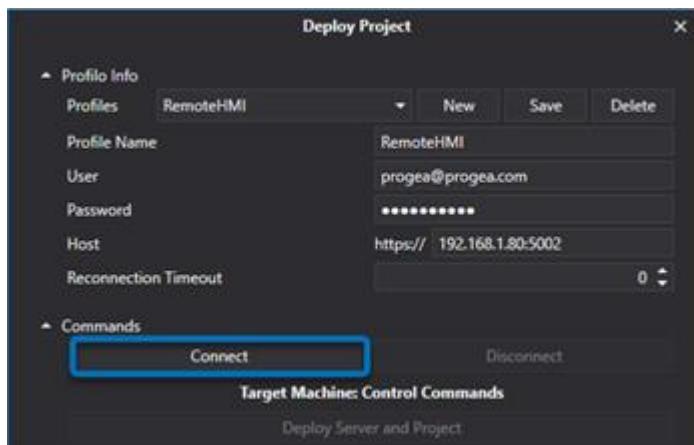
In order to proceed with activating a WebHMI runtime license you will need to have the license file with the "mlztrack" extension available and the Deploy Server installed on the HMI as indicated in the Movicon.NExT help online chapter: WebHMI.

Once the Deploy Server is running, go to the Movicon.NExT editor and proceed as follows;

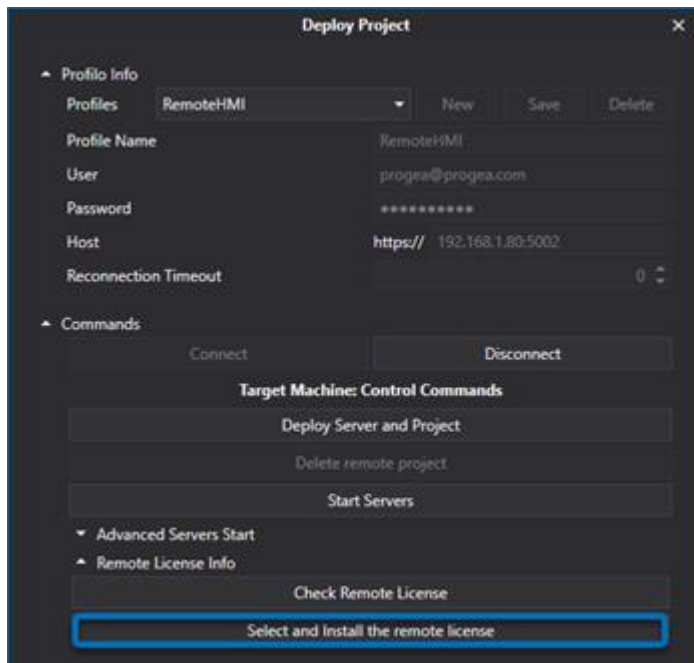
1. Select the icon to open the Deploy Tool



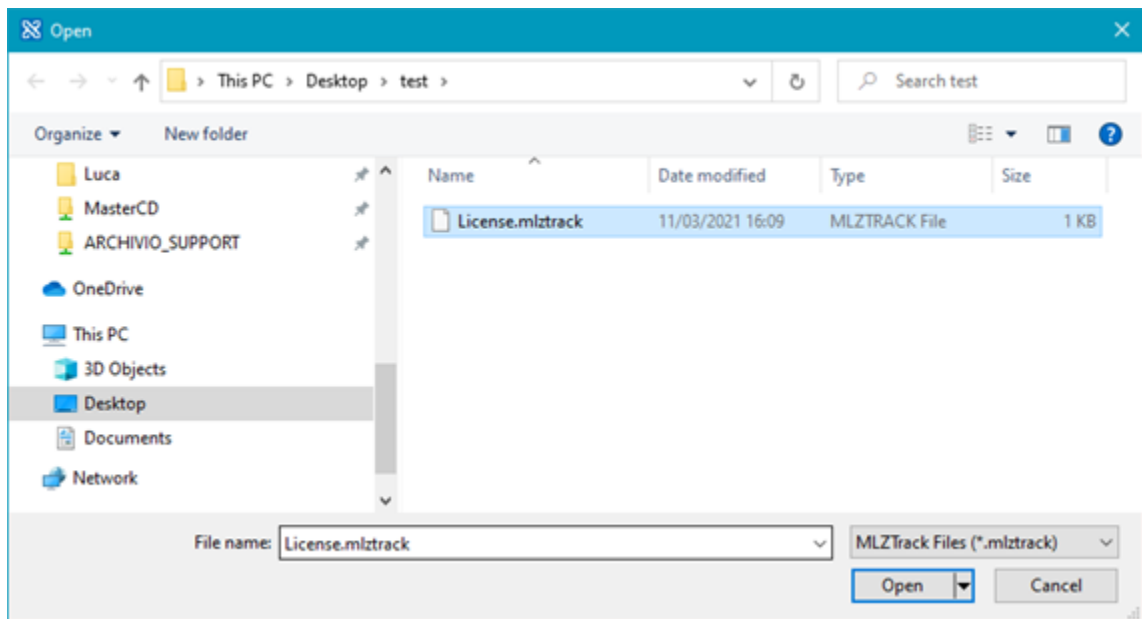
2. Once the Deploy Project window has opened connect to the remote machine's Deploy Server by configuring the connection parameters and selecting Connect:



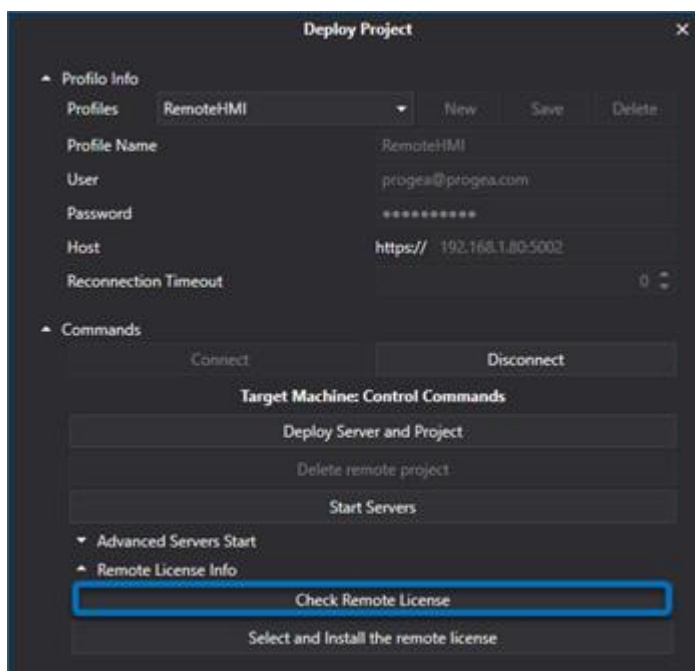
3. Once connected, you will be able to use: "Select and Install the remote license"



4. This will open another window with the license file with the "mlztrack" extension for you to select:



Confirmation will be given once the license has been installed.  
You will be able to check the contents of the installed license by using the “Check Remote License” option:

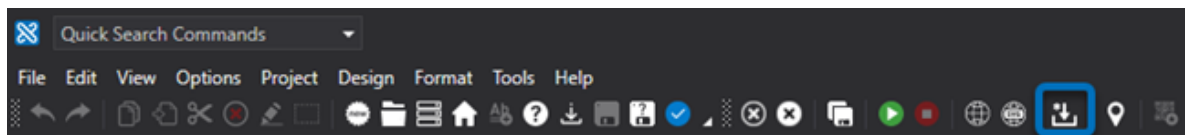


### How to retrieve the Site Code of a remote WebHMI Runtime

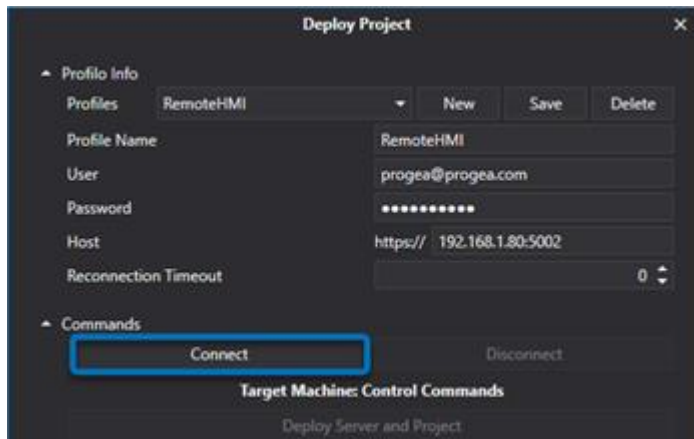
To retrieve the site code of a WebHMI, you will need to install the Deploy Server on the HMI as indicated in the Movicon.NExT help online chapter: WebHMI.

Once the Deploy Server is running, go to the Movicon.NExT editor:

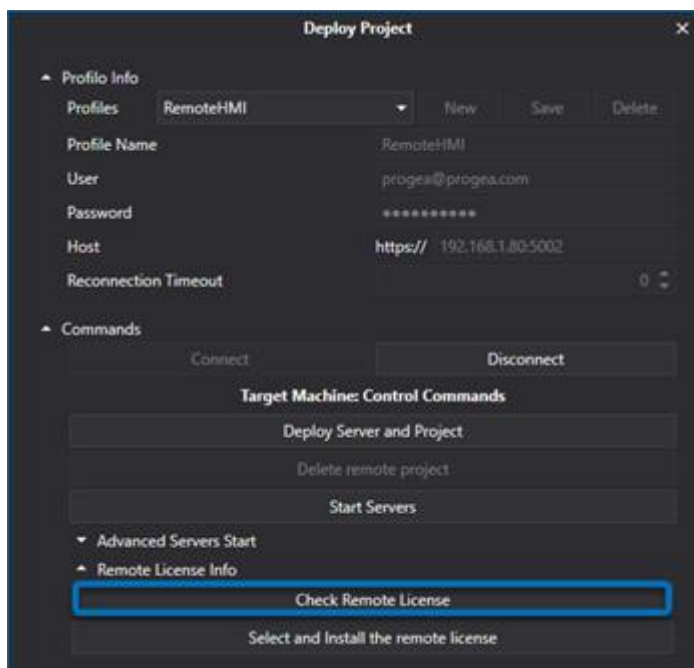
1. Select the icon to open the Deploy Tool



2. Once the Deploy Project window has opened, connect to the remote machine's Deploy Server by configuring the connection parameters and selecting Connect:

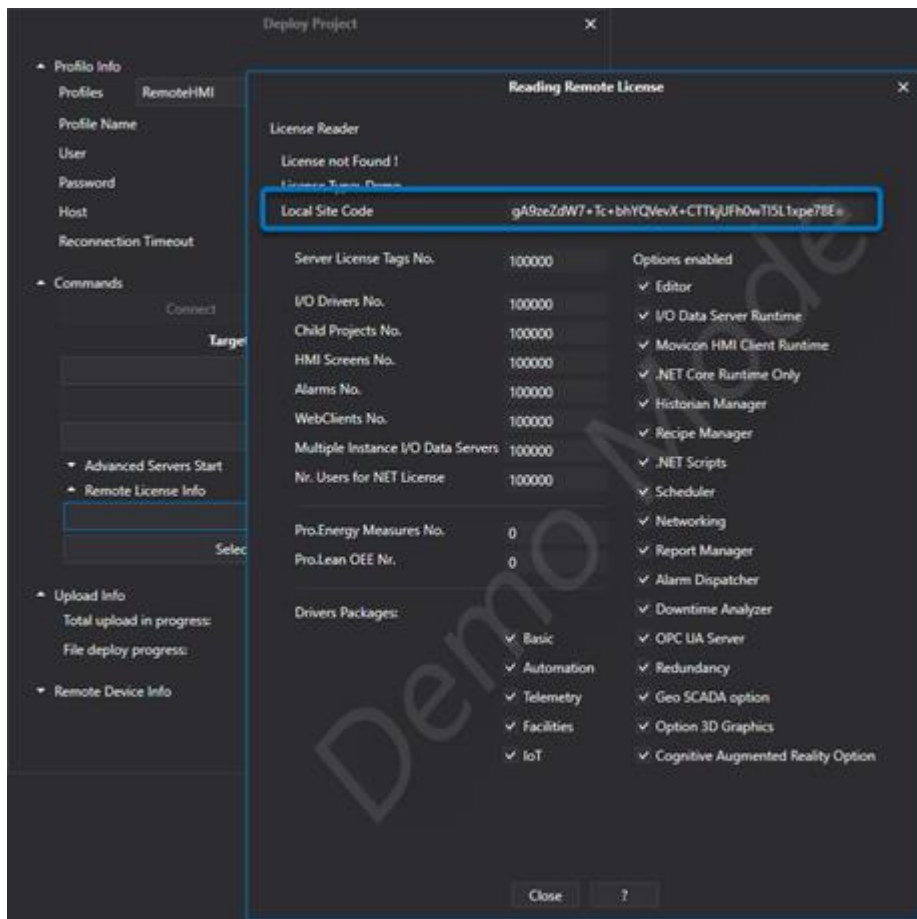


3. Once connected, you will be able to use: "Select and Install the remote license"



4. Another window will open showing the current license in use in the machine (It will be running in Demo mode on the remote WebHMI Runtime). You can then copy the "Local Site Code" string which will help generate the license.





## 16.10. Cognitive Augmented Reality

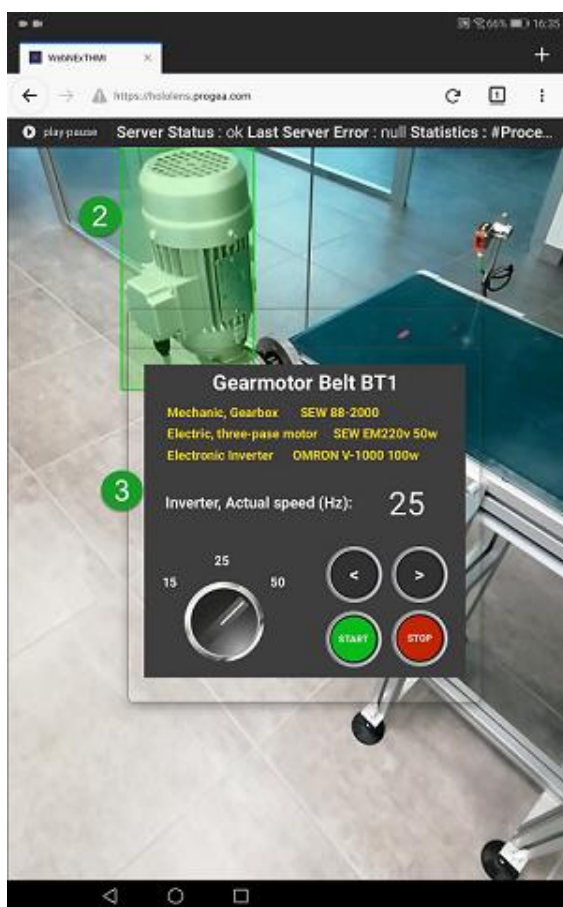
### 16.10.1. Introduction

Movicon.NExT has been designed to allow users to take the advantage of the most advanced Machine Learning technologies aimed at detecting and recognizing objects in the field to immediately visualize their dynamic real-time information for operator scrutiny.

This technology allows operators to use their mobile device's integrated video camera or augmented reality devices to combine cognitive system information with the Movicon.NExT project by using any HTML5 browser with the WebHMI technology. This document describes the technologies on which the platform is based and how to use and configure them.

### Cognitive Augmented Reality Applications with Movicon.NExT

The scope of this solution, which is integrated in Movicon.NExT, is to facilitate the creation of HMI systems with the capability to use the camera eye lens to recognize previously learned elements or components in the plant. Due to the real-time recognition of objects in the surrounding environment, operators can automatically receive the relative HMI real-time information through automatic pop-up screens previously defined in the project in relation to the targeted context. The MI screens can in turn contain dynamic project information by displaying variables in animated screen format and allow the use of interactive objects to invoke commands.

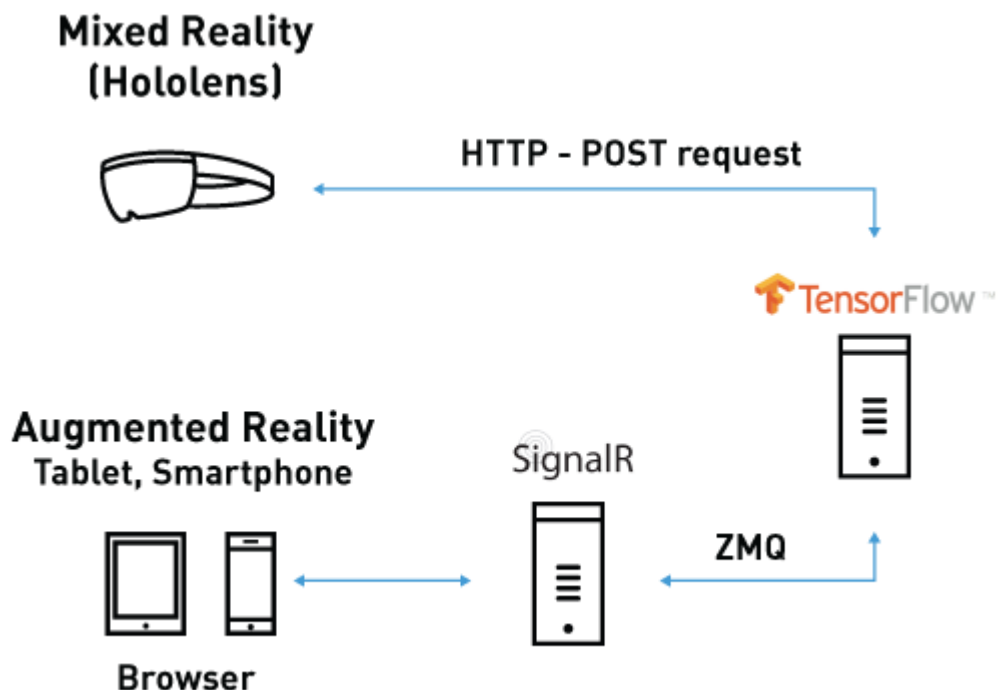


For example:

1. Once the browser has been opened and the video camera activated, the operator will be able to frame their surrounding environment
2. The system will analyze the streamed video in order to recognize the objects it has been trained to identify and indicate them on screen enclosed within a boundary box;
3. Once the target object has been recognized and tracked, a pop-up window will show with its corresponding information. This gives the operator direct access to all the information typical of an HMI, such as displayed digital data, alarms and historical information, along with the ability to execute operational commands.

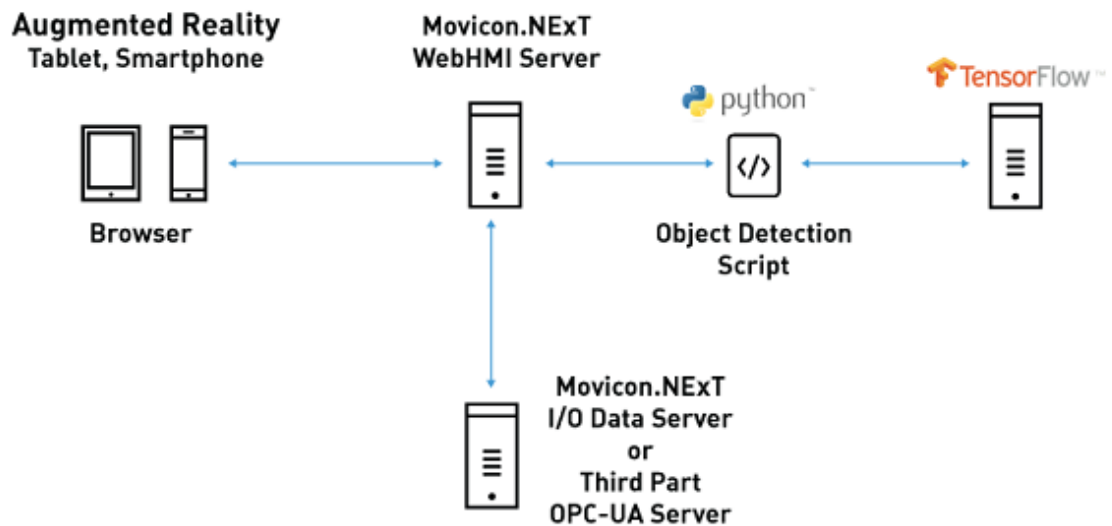
### The difference between Augmented Reality and Mixed Reality

Augmented Reality refers to the visualization of real-world images captured by the camera in devices that are overlaid with additional dynamic information processed by the Artificial Intelligence systems. Augmented Reality is supported by devices equipped with rear cameras such as tablets and smartphones. Mixed Reality refers to the management of devices that offer a more immersive experience allowing operators to interact with and manipulate both physical and virtual items and environments using wearable devices such as the Microsoft HoloLens with next-generation sensing and imaging technologies. These holographic devices display additional dynamic information through the holographic visualization of images displayed in the surrounding space using appropriated holographic displays. From a functional standpoint, there are no differences in use between the two. However, from a technical standpoint, the process technologies are slightly different, as illustrated in the following diagram:



### 16.10.2. Cognitive Augmented Reality System Architecture

The Cognitive Augmented Reality system architecture encompasses the following independent software components:



- One or more mobile devices (Tablet, Smartphone) equipped with a rear camera that uses a browser to stream video to local websites exposed by the WebHMI Server.



The use of wearable devices, such as the smartglass and HoloLens, is limited because these provide app browsers that can be opened at full screen and which remain open in front of the person's line of vision while moving in the space around them.

This function, for example, is available in HoloLens2 and called "Follow me":  
<https://docs.microsoft.com/en-us/hololens/holographic-home>

- WebHMI Server, which receives the video streaming from browsers and sends it to the TensorFlow recognition engine by means of Python script. The results of the TensorFlow process are used open pop-up screens with the same as the recognized objects. The WebHMI Server interfaces with the Movicon.NExT I/O Data Server to graphically represent or control the variables exchanged with the field.
- "Object Detection" Python script has been provided to allow communications between the WebHMI Server and TensorFlow Server.
- A Server TensorFlow whose job is to process video streaming, identity recognized elements (previously learned through the learning phase), and respond with the name and coordinates of the recognized elements.
- A TensorFlow Server which handles the video streaming process by identifying the known objects (previously learned in the training phase) and responding with their names and coordinates when recognized. TensorFlow is an open source software library for machine learning, which provides

tested and optimized modules that are useful for creating algorithms for different types of perceptual tasks such as object recognition.

- I/O Data Server, which deals with field communications by means of using communication drivers and managing historical data and alarms. Other third party OPC-UA Servers can be used as an alternative to using the Movicon.NExT I/O Data Server.



The software components illustrated above can all be installed on one machine or on different machines.

### Hardware and Software Requirements

The TensorFlow recognition engine requires a high calculation capacity regarding the video side in particular.

- CPU: Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz
- RAM: 16 GB
- HD: disco SSD
- Video Card: GeForce GTX 1080 or equivalent

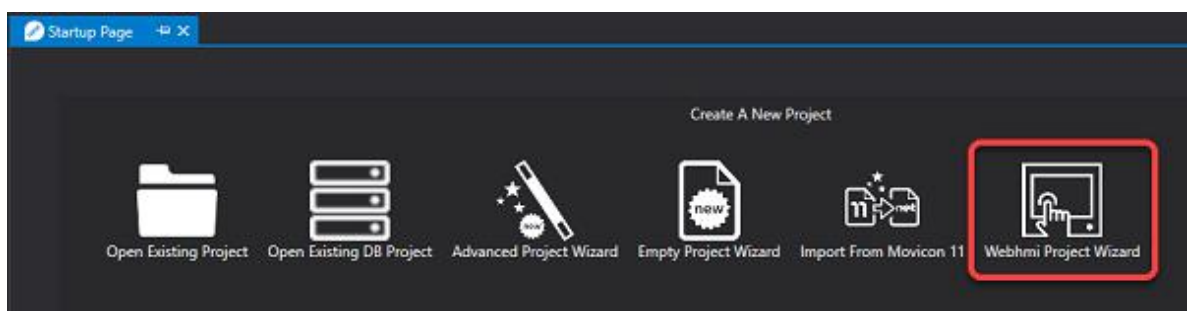
The recommended software requirements are:

- Operating System: Ubuntu 16.04 LTS or higher
- TensorFlow 1.8.0
- Python 3.5.2

### 16.10.3. Project and Movicon.NExT WebHMI Configuration

To develop a WebHMI project as a Cognitive Augmented Reality HMI application simply follow the procedures described below:

1. Create a new Movicon.NExT project by making sure you choose the “WebHMI Project” model in the Startup Page.



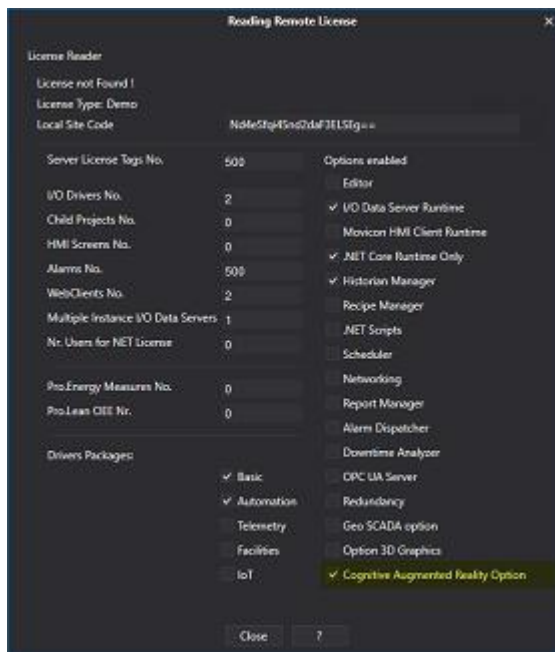
2. Create the screens to be displayed as object recognition pop-up windows. The screens can be created as normal screens and can contain any display object and/or control available in Movicon.NExT WebHMI. The name of each of these screens should have the same name of the object (or Model) recognized by TensorFlow.

3. The WebHMI Server should be configured in a way that no project screen showing live images from the devices video camera display to allow pop-up screen to show when called based on the objects recognized by the TensorFlow. Simply edit the appsettings.json file included with the zip "C:\Program Files\Progea\Movicon.NExT 4.1\WebHMIserver.zip" zip packet by setting the "ForceMainPageAR" property to true.



```
1 {
2   "Logging": {
3     "IncludeScopes": false,
4     "LogLevel": {
5       "Default": "Debug",
6       "System": "Information",
7       "Microsoft": "Information",
8       "Microsoft.Hosting.Lifetime": "Information"
9     }
10  },
11
12  "AllowedHosts": "*",
13
14  "RaygunSettings": {
15    "ApiKey": "FGaR3Z3O0inisTemfkZwVg=="
16  },
17
18  "WebNExTHMISettings": {
19    "Project": "",
20    "LoginRequired": false,
21    "ShowDebugWindow": false,
22    "ForceMainPageAR": true,
23    "ARMappingProject": null,
24    "HttpListeningPort": 5000,
25    "HttpsListeningPort": 5001
26  }
27 }
```

4. To transfer all the Movicon.NExT components to the Linux Machine you will need to launch the project into Runtime: WebHMI Runtime, I/O Data Server and Project. Please follow the indications in the Help Online's "WebHMI Runtime Linux (x64)" chapter to do this successfully.
5. Make sure that the Augmented Reality option on the Movicon license, installed on the machine where the WebHMI Server and I/O Data Server will be running, has been activated. To verify/install the license on the virtual machine, use the Deploy Tool and "Check Remote License" and "Select and install the Remote License" commands.

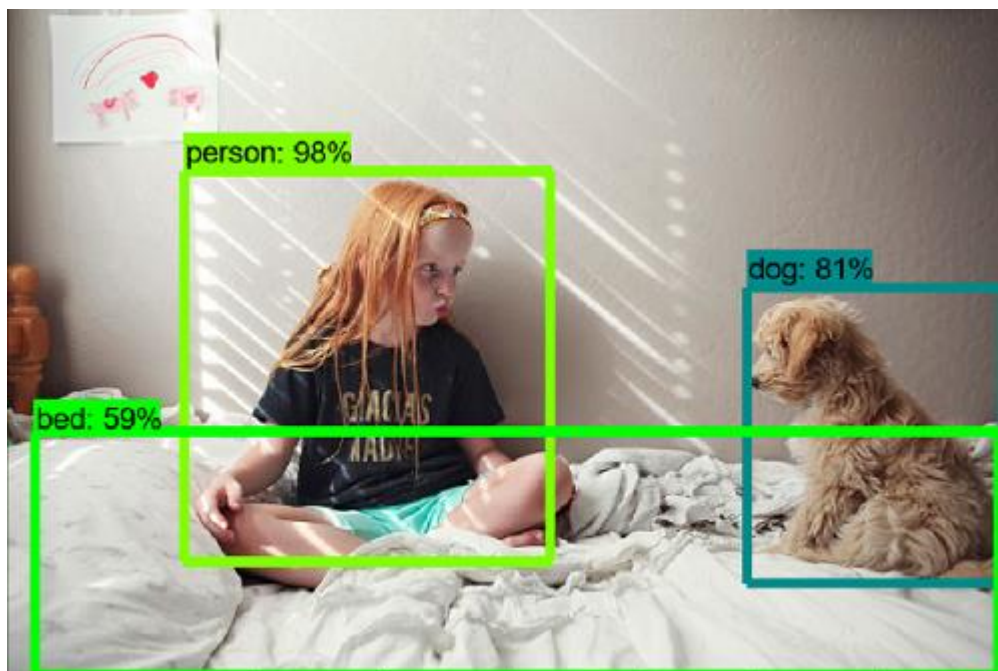


#### 16.10.4. TensorFlow Installation and Configuration

TensorFlow is an Open Source software that can be downloaded at:

<https://www.tensorflow.org/install>

Once installed, TensorFlow has to be trained using a "Training" procedure to know which objects are to be recognized and which will be named "Models". The system offers the possibility to use read-to-use available models available for recognizing objects and commonly used subjects. In this case, TensorFlow offers the "Common Object in Context" (COCO) system, which consists of a database containing 90 object classes (Models) with over 200 thousand images.



Conversely, you can train the system to recognize your own custom models. In order to do, this you will need to use the Training procedure to add your models by creating the images of the objects to be



recognized. The models should be based on a series of images taken in different positions, angles and light modes, so that the TensorFlow analysis engine can recognize the desired object in all circumstances and angles. The greater the number of shots taken, the better TensorFlow will be able to perform a definite object recognition in all conditions.

## Training Procedure

The TensorFlow Training involves the following steps:

1. Download: a load of images relating to the objects to be recognized are loaded to the system.
2. Labelling: each image is labelled using a python script
3. Separation: the images are separated by those used for Training and those used for Testing.
4. Conversion: the files of the images are converted into the formats that are used by TensorFlow
5. Configuration: a series of system configuration files are set.
6. Start Training: the actual Training procedure starts by providing the system with the converted image files and configuration files.

### Step 1: Download

The more images (jpg or png) you produce for each object, the more effective recognition will be. We suggest you take different photos of each object with different depth, inclinations, light and light tones.

- Remember that the greater the number of images downloaded, the greater the model performance will be in recognizing the objects present!
- SSD: 300x300
- FastRCNN: 600x600

The images can be scaled automatically by using the "resize.py" python script (<https://pypi.org/project/python-resize-image/>). There is also a "rename.py" (<https://pypi.org/project/py-rename/>) script that allows all the images in the folder to be renamed starting from the initial value to be established, up to the last image in ascending order. When terminating this phase, we should have created a folder containing a certain number of images relating to the object to be recognized, all of more or less the same size.

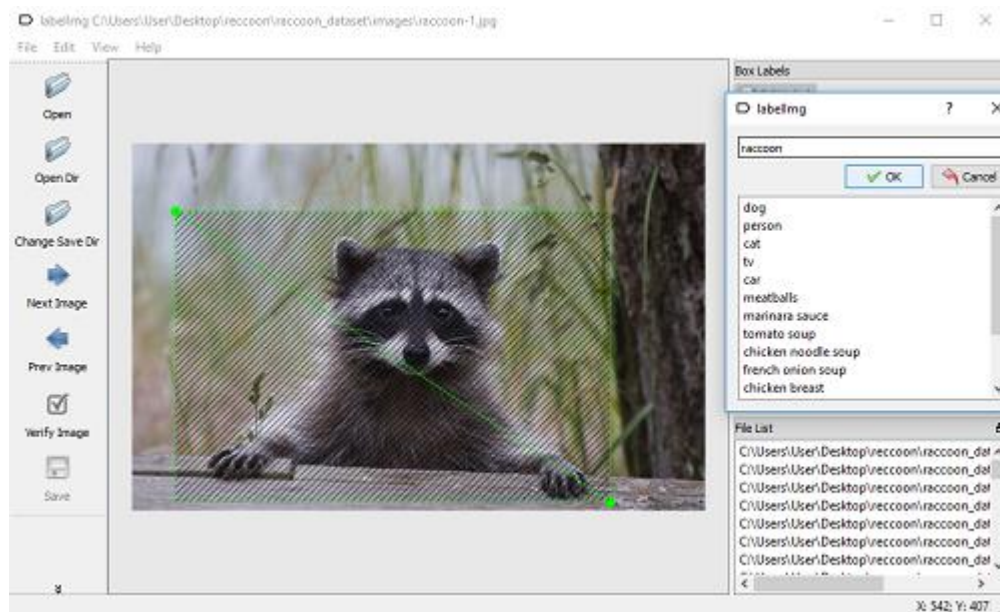


before going ahead with the next phase to label each individual image, double check that they are all in the right size and format.

### Fase 2: Labelling

By using TensorFlow, you will now have to label each image, one by one, tracing by hand bounding boxes to identify and circumscribe the object.





Use a software for labelling the images, such as LabelImg (<https://pypi.org/project/labelimg/>) which should allow you to:

- Download images.
- Draw a bounding box around the object.
- Assign a class to the bounding box.

The “labelIMG.py” Python Script allows you to:

- Select the folder containing the images to be labelled.
- Select the folder where the generated .xml files will be saved.
- Define the classes to be assigned.

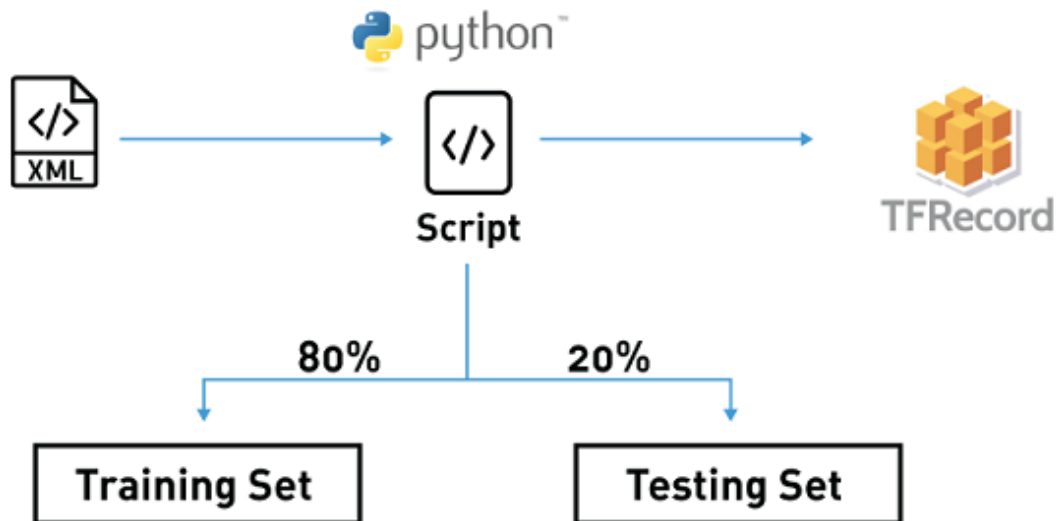
Some handy shortcuts:

- **w** = to draw the bounding boxes
- **ctrl + s** = to save the .xml file
- **d** = to go to the next image
- **a** = to go back to the previous image

The software generates an XML file for each image.

### Fase 3: Separation

Before converting the files, you will need to choose a percentage of the XML files (around 20%) to use for validation while the remaining (around 80%) will be used for the Training phase.



#### Fase 4: Conversion

After having separated the XML files in two separate folders, you can now proceed with converting them by using the "xml\_to\_csv.py" and "generate\_tfrecord.py" ([https://github.com/datitran/raccoon\\_dataset](https://github.com/datitran/raccoon_dataset)) scripts. The first of the two above mentioned scripts can be run in any folder by simply adjusting its internal parameters:

- image\_path = path where to find the .xml files
- xml\_df.to\_csv = path where to save the .csv files

The second script must be run in /research. In particular, you should insert the below listed in this folder (or in a new 'data' folder):

- the .csv files to be converted to TFRecords,
- the folder with all the images
- the script to be used for the conversion.

Before running the script, you will need to change the code inside that corresponds to the "label\_map" with the one made available.

```

def class_text_to_int(row_label):
    if row_label == 'Emergency':
        return 1
    elif row_label == 'Manometer':
        return 2
    else:
        None
  
```

The command to execute is:

```

python3 generate_tfrecord.py --csv_input=data/train_labels.csv --output_path=train.record
python3 generate_tfrecord.py --csv_input=data/test_labels.csv --output_path=test.record
  
```

By the end of this step we should have: two .record files which are "train.record" and "test.record"

#### Fase 5: Configuration

This step involves preparing the configuration file need for the system to function correctly.



THE NAME ASSIGNED TO THE MODEL IN THE CONFIGURATION FILE SHOULD

CORRESPOND TO THE NAME OF THE CONTEXTUAL SCREEN THAT MOVICON WILL DISPLAY UPON RECOGNITION!

**1. Pipeline Configuration File:**

- Model type to use for training
- Parameters to use for the model training
- Testing metrics
- Which dataset to train the model on
- Which dataset to test the model on

**2. Label Map: contains the correspondence between the identifier and the class name. In order to compile correctly, the first ID should be 1.**

**3. Pre-Trained Model (optional):** we recommend that you use a pre-trained model to facilitate training convergence.

- Pipeline configuration file: a file in which a series of parameters need to be configured, while the others can be left set for default. Those that you will need to configure are:
  - Number of classes
  - The label map's path
  - Which dataset to train the model on (train.record path)
  - Which dataset to evaluate the model on (test.record path)
  - Use a pre-trained model's weights to converge earlier

The various default fields include a very useful one called `data_augmentation_options`, whose parameters can be set to make the available dataset more fine-tuned.

1.

- **Label map:** this shows the correspondence between an identifier and the class name. This map should always start with an ID equal to 1.

```
item {  
  id: 1  
  name: 'Emergency'  
}  
item {  
  id: 1  
  name: 'Manometer'  
}
```

- **Starter Pre-trained Model: this is optional.** We advise you to use it because it greatly simplifies the training convergence. A range of pre-trained models are available for you to choose from depending on whether you want faster recognition or better accuracy. (`ssd_mobilenet_v1_coco` is used for faster recognition).

Use this link to find others:

[https://github.com/tensorflow/models/blob/master/research/object\\_detection/g3doc/detection\\_model\\_zoo.md](https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md)

## Fase 6: Start Training

The last step involves starting up the Training procedure.

The Training and Testing procedures can be performed by using two different methodologies:

- Cloud: uses the Google Cloud Services characterized by high performance computing and charges a fee to use them. In this case, you will need to load the corresponding files in a Google Cloud Bucket.
- Local: everything is performed locally, therefore the computing capacity depends on the HW available. The computational calculation for local training can be performed by means of using

the CPU or GPU. Computing performed with the GPU is faster compared to that of a CPU but requires a video card with at least 2GB of memory.

To launch the training operations use this command:

```
# From the tensorflow/models/research/ directory
python object_detection/train.py \
  --logtostderr \
  --pipeline_config_path='models/ssd_mobilenet_v1_pets.config' \
  --train_dir='checkpoints'
```

To launch the testing operations use this command:

```
# From the tensorflow/models/research/ directory
python object_detection/eval.py \
  --logtostderr \
  --pipeline_config_path='models/ssd_mobilenet_v1_pets.config' \
  --checkpoint_dir='checkpoints' \
  --eval_dir='evaluations'
```



NB: modify the .config with  
eval\_config: {  
 num\_examples: 10 (file number to validate)  
 num\_visualizations: 10  
 eval\_interval\_secs: 0  
}

You can also launch the Tensorboard platform to visualize the training and evaluation progress. The training process will continue until you stop it.

```
tensorboard --logdir='checkpoints'
tensorboard --logdir='evaluations'
```

Once the training process has finished, you will need to export the trained model in one file (Tensorflow graph proto) so that it can be used for detecting objects.

The command used to export the model starting from the modelcheck created during the training process is: models.md

[https://github.com/tensorflow/models/blob/master/research/object\\_detection/g3doc/exporting\\_models.md](https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/exporting_models.md)

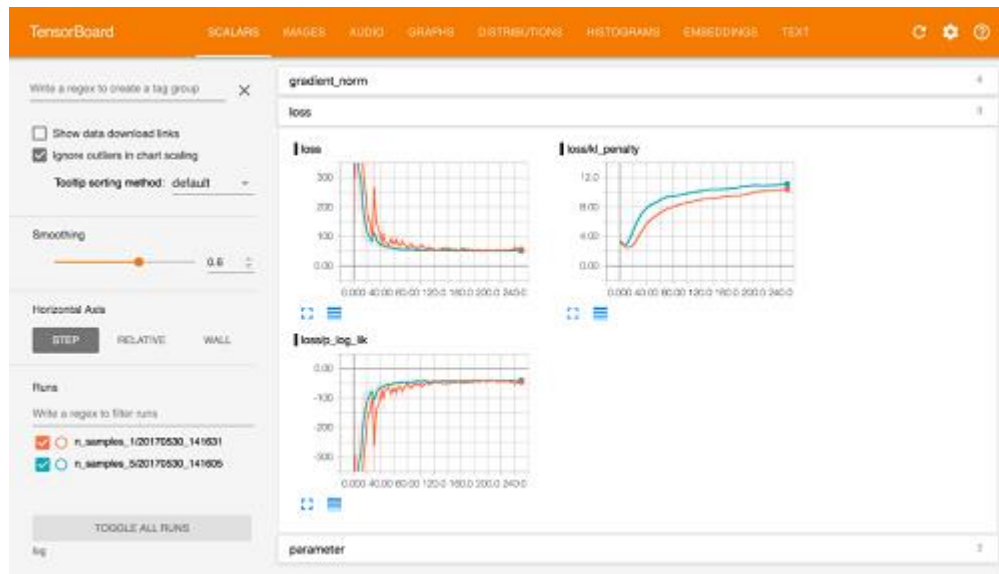
```
python3 object_detection/export_inference_graph.py --input_type=image_tensor --
pipeline_config_path=models/ssd_mobilenet_v1_pets.config --
trained_checkpoint_prefix=checkpoints/model.ckpt-10405 --
output_directory=output_inference_graph
```

Large amounts of data are needed to make the model robust and performing. We have only talked about the recognition of one individual class. Model training can obviously be applied to a greater number of classes which will take longer to train and need more data. The loaded images must contain the classes to be recognized, either individually or together with other classes, so that the machine can learn to recognize the object even when other objects are present without confusing the system.

### How long does it take to train a model?

- It will continue until the operator interrupts it.
- It should be interrupted when the error continues to persist
- It should be interrupted when accuracy does not improve.

The TensorBoard function, provided with TensorFlow, can be used to monitor behavior while running the training and testing procedures.



Please refer to the TensorFlow documentation for more detailed and up-to-date information. The indications given here are not intended to be exhaustive and are only an outline on how to configure and use the system.

TensorFlow is an Open Source software developed by Google Brain. Therefore, its use is only described here in combination with Movicon.NexT WebHMI by fully acknowledging the intellectual property and copyrights of the legitimate owners.

### 16.10.5. Python Script Installation and Configuration

The two platforms: WebHMI and TensorFlow, use the "object\_detection\_signalR\_multiThread\_REQRES.py" Python script which have been developed to communicate with each other. This script allows TensorFlow to receive video streams and allows WebHMI to receive the detected model's name, that will then correspond to the screen to be displayed in the project.

#### Script Installation

The "object\_detection\_signalR\_multiThread\_REQRES.py" script can be downloaded from this link:

[Download]

and should be copied in the TensorFlow folder in path:

...tensorflow/models/research/object\_detection/object\_detection\_signalR\_multiThread\_REQRES.py

You can modify/configure the script as needed in your architecture. The parameters that can be used to do this are shown in the image below:

```

# Port used by Script to listen for WebHMI connection
BIND_PORT = '8080'

# IP Address and Port to connect to WebHMI service
CONNECT_TO_IP = '127.0.0.1'
CONNECT_TO_PORT = '8181'

# What model to download.
MODEL_NAME = 'output_inference_graph_rcnn'

# Path to frozen detection graph.
# This is the actual model that is used for the object detection.
PATH_TO_CKPT = MODEL_NAME + '/frozen_inference_graph.pb'

# List of the strings that is used to add correct label for each box.
PATH_TO_LABELS = os.path.join('data', 'rull_label.pbtxt')

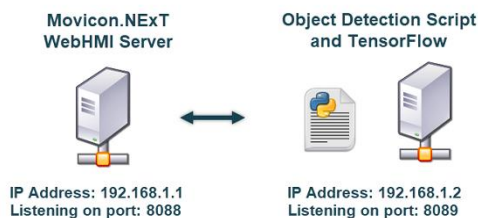
# Number of classes to be recognized
NUM_CLASSES = 5

```

In particular, the BIND\_PORT, CONNECT\_TO\_IP and CONNECT\_TO\_PORT default models need to be configured so that the WebHMI service and the script (and as a consequence also TensorFlow) are running in the same PC.

In this condition, the WebHMI service is supposed to listen on the port 8181, while the script listens on the port 8080.

For instance, if the WebHMI service happens to be running on a different PC and not on the same PC where the Tensorflow/script is running, you should proceed as follows:



In a condition like the one shown in the above image, you should implement the following modifications:

1. In the "object\_detection\_signalR\_multiThread\_REQRES.py" script set:
  - BIND\_PORT = '8089'
  - CONNECT\_TO\_IP = '192.168.1.1'
  - CONNECT\_TO\_PORT = '8088'

as shown in the following image:

```

# Port used by Script to listen for WebHMI connection
BIND_PORT = '8089'

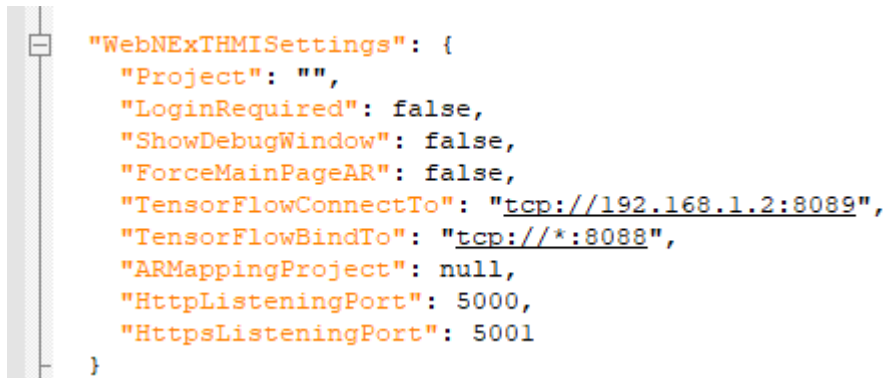
# IP Address and Port to connect to WebHMI service
CONNECT_TO_IP = '192.168.1.1'
CONNECT_TO_PORT = '8088'

```

2. The following parameters should be added to "appsettings.json" file residing in the WebHMI service's root folder:

- "TensorFlowConnectTo": "tcp://192.168.1.2:8089",
- "TensorFlowBindTo": "tcp://\*:8088",

as shown in the image below:

A screenshot of a code editor with a light gray background. On the left, there is a vertical scrollbar and a small square icon. The code is written in a monospaced font with orange and black text. It defines a configuration object named "WebNExTHMISettings" with various properties including "Project", "LoginRequired", "ShowDebugWindow", "ForceMainPageAR", "TensorFlowConnectTo", "TensorFlowBindTo", "ARMappingProject", "HttpListeningPort", and "HttpsListeningPort".

```
"WebNExTHMISettings": {  
  "Project": "",  
  "LoginRequired": false,  
  "ShowDebugWindow": false,  
  "ForceMainPageAR": false,  
  "TensorFlowConnectTo": "tcp://192.168.1.2:8089",  
  "TensorFlowBindTo": "tcp://*:8088",  
  "ARMappingProject": null,  
  "HttpListeningPort": 5000,  
  "HttpsListeningPort": 5001  
}
```





# 17. Runtime Manager

## 17.1. Automatic project startup from command line

Runtime can be started up by using a command line without needing to open the project with the editor first. The command line, which can be used within a shortcut or a batch file, is composed of the following parts:

- "MoviconNExTRT.exe" executable path
- Parameters:
  - file
  - start
  - client
  - screen
  - nosplash



The parameters are Case Sensitive!



We advise you to use of the double speech marks " " in defining the path to avoid problems with spaces in the path.

### "MoviconNExTRT.exe" executable path

Indicates the MoviconNExTRT.exe absolute executable path, for example: "C:\Program Files\Progea\Movicon.NExT x.x\MoviconNExTRT.exe"

### "-file" Parameter

This parameter allows the you to indicate the path of the project to be launched.

- When the project has been save on file, simply indicate the absolute path (local or in net) of the project file with the UFProject extension.  
Local path example: -file="E:\Projects\Project1\Project1.UFProject"  
Network path example: -file="\\Hostname\Projects\Project1\Project1.UFProject"
- When the project is on Datasource, you should use the connection string to the database instance in which the project is saved.  
Example: -file="XpoProvider=MSSqlServer;data source=DB-SERVER\SQLEXPRESS;integrated security=SSPI;initial catalog=Project1"

In this case, the project is saved on the "Project1" DB of the "DB-SERVER\SQLEXPRESS" instance and access to the DB is by means of integrated Windows Login..

It is also possible to use SQL login by changing the connection string in the following way:

Example: -file="XpoProvider=MSSqlServer;data source=DB-SERVER\SQLEXPRESS;user id=User;password=Pass;initial catalog=Project1;Persist Security Info=false"

### "-start" Parameter

Indicates that runtime is to be started up.

### **"-client" Parameter**

Indicates that only the project's Client side is to be started up.



In this case, if the NExT I/O Data Server is not local, make sure that the project has been configured to permit Client connection to a remote Server as described in "Client connections to Server"

### **"-screen" Parameter**

Indicates the name of the screen to be loaded as Client startup.

The name should be preceded by "Screen\" and it should indicate the ".xaml" extension.

Example: -screen="Screen\Screen1.xaml"

<INFO>When this parameter is this parameter is omitted, the TilePage, or the MainScreen configured in the Project's properties, will be loaded.

### **"-nosplash" Parameter**

Avoids displaying the Movicon.NExT splash screen at Runtime startup.

### **Command Line examples**

Here are some examples of command lines:

1. Server and Client startup with local project: "C:\Program Files\Progea\Movicon.NExT x.x\MoviconNExTRT.exe" -file="E:\Projects\Project1\Project1.UFProject" -start
2. Client only startup with local project: "C:\Program Files\Progea\Movicon.NExT x.x\MoviconNExTRT.exe" -file="E:\Projects\Project1\Project1.UFProject" -start -client
3. Client only startup with local project with a different non default startup page: "C:\Program Files\Progea\Movicon.NExT x.x\MoviconNExTRT.exe" -file="E:\Projects\Project1\Project1.UFProject" -start -client -screen="Screen\Screen1.xaml"
4. Client only startup with local project without splash screen: "C:\Program Files\Progea\Movicon.NExT x.x\MoviconNExTRT.exe" -file="E:\Projects\Project1\Project1.UFProject" -start -client -nosplash

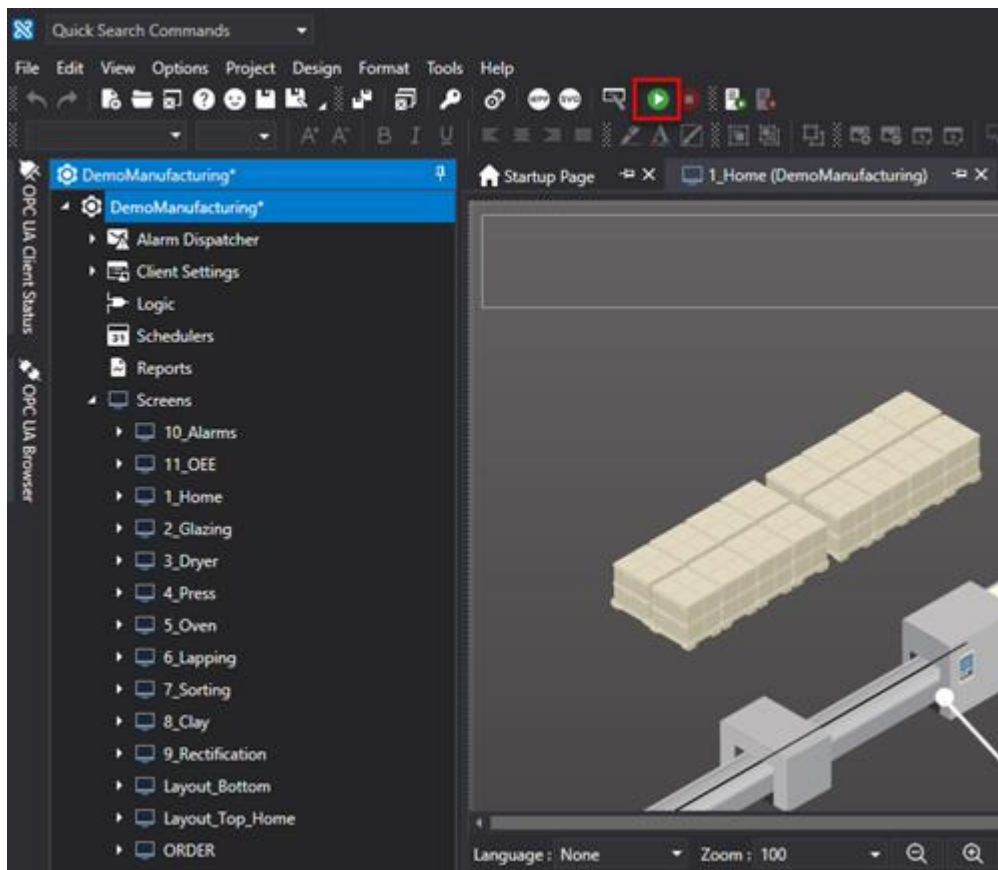
## **17.2. Starting Project up in Editor**

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



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

You can launch entire project into runtime from the workspace by using the "Start Runtime" command available from the Project's Toolbar or the File > Start Runtime menu.



*The Start Runtime command starts Project up in runtime*

When the "Start Runtime" command is activated generically, the platform will startup both the Server and the Client module. The modules, however, are independent and therefore the project can be set to startup in "Server Only" or "Client Only" mode as well.

## Startup Project Type

By means of using the "**Startup Page Type**" property in the project's properties window, you will be able to select the project's startup page display type when launched into Runtime (Start Runtime command). The modes available are:

- **Main Screen:** This startup page type will be one of the screens created and inserted in the project. The screen you wish to use as the startup screen can be selected from the "MainScreen" property. If this property is left empty, Movicon will search for a screen with the name of "Main". If it is found, Movicon will start it up otherwise it will startup with the first project screen in alphabetical order.
- **Tile Page:** This startup page type will automatically show with a 'Tile' interface similar to that of Windows 8. It will be composed of the same number of square tiles as there are screens in the project. Each square tile will report the name of the screen and its preview. Clicking on the square tile will open the corresponding screen.
- **Geo Page:** This startup page will show a geographic map on which the locations of the project resources are shown. In this case, the project is usually composed of one or more child projects dislocated in different geographical areas. A click on one of the sensitive sites will open the corresponding project.

## 17.3. ClickOnce Client Startup

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

### Client ClickOnce Installation

The Movicon.NExT ClickOnce version is a Client version only. This means that the Platform NExT Editor and I/O Data Server components will not be installed.

The installation procedure is done by simply clicking on a link provided on the web page exposed by a Server that will then be the Server used for updating.

You can get access to the Movicon.NExT ClickOnce Setup at the following link:

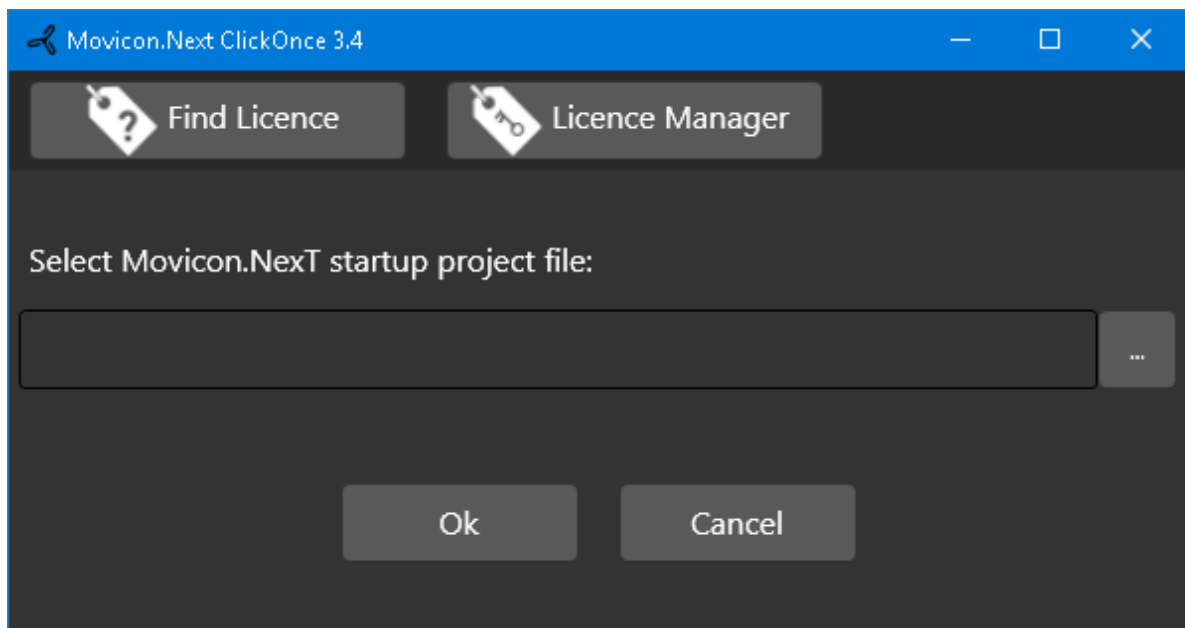
<http://www.movicon.info/download/setup/PlatformNextWebSetup3.4/ClickOnce/publish.htm>

You can also request a custom version of the Movicon.NExT ClickOnce to install in your own local Server.

### ClickOnce Client Startup

Once the Setup has terminated, the client will automatically display a setup window where you can:

- Select the Movicon.NExT project to be launched at each execution
- Verify license status
- Manage license activation/deactivation

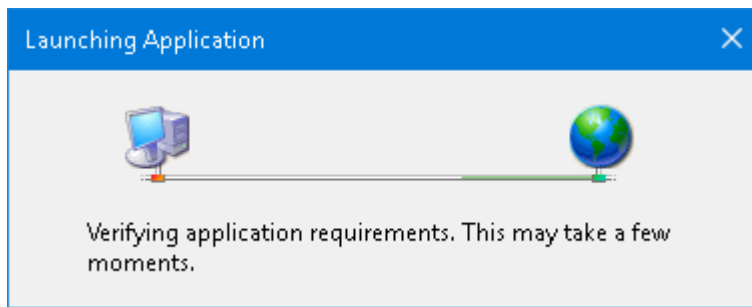


The project will then startup in 'Client Only' mode as it does when using the "--start -client" parameters for starting up the Client from command line.

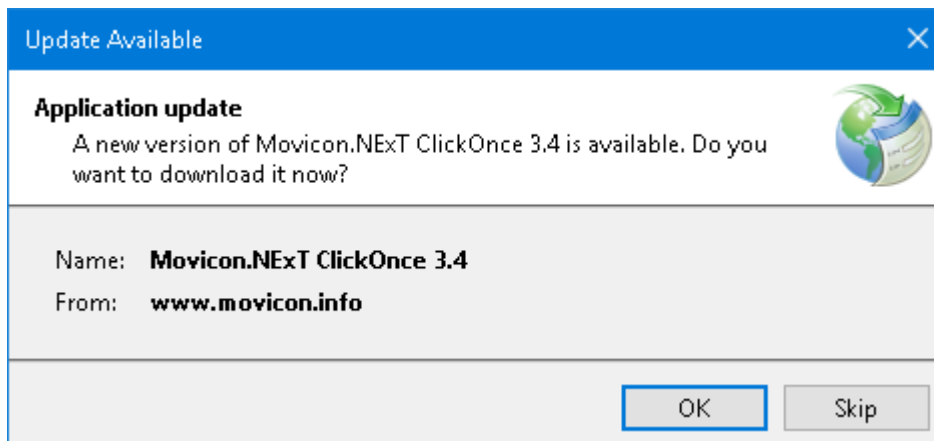
To reopen the setup window, in order to change which project to run for example, simply launch Movicon.NExT with the CTRL key pressed down.

### ClickOnce Client Update

Each Movicon.NExT ClickOnce application launch interrogates the Server in which it has been installed to check for updates.



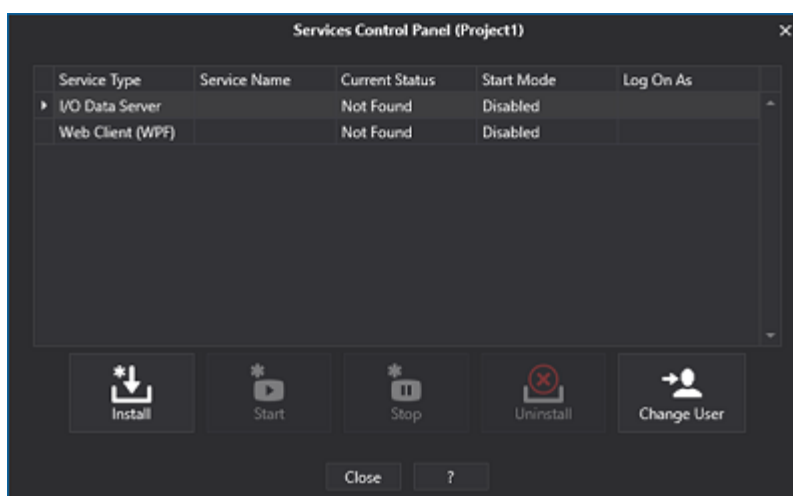
If any updates are available, another window will show asking you to update or skip this procedure until later.



## 17.4. Starting up as Service

The Movicon.NExT Data Server can also be installed as a **Windows Service**. The Windows Services allow applications that support this functionality to run automatically at startup independently from the local operating system or Domain user authentication.

In the Options menu you will find the "Service Control Panel" command to access the Tool used for installing and managing the project's components as Windows Service. By using the window shown below, you will also be able to install and manage the I/O Data Server as service.



### How to install and configure the Service

In order to make the Server project startup as Windows Server you first need to install it in the Windows Services and configure any startup authentication and security procedures.



The Windows Services are specific applications that are run within the context of the Operating System independently or potentially independent from the user interface and Windows Log On. Therefore you need to consider that a Service is a Server application that can be run in the local context or with local or domain security policies as we will see ahead.

To install your Server project as a Service, use the "Install Service" command from the Project> I / O Data Server> Services Management menu.

This window is used to enter the credentials that will be used by the Service. When no credentials are entered here, the Service will be run on the 'Local' context. Otherwise, the desired credentials are entered as described below.

- When the project is installed as service, it is identified with the same project name among the Windows Services so that other projects can be installed as service at the same time.

## Service User Credentials and Security

As mentioned beforehand, installing the Server project as Service is done using the Service Manager Tool from the Movicon.NExT editor.

When installing the service leaving the credential fields empty, the service will be run only within the "local" context (Local System).

If you wish to use Windows Security instead, you will need to use the access credentials for running the service.

Based on the credential of the user name, the server can be run as:

- Local User Password
- Domain User

### Local User Password

When entering a User Name and Password that have not been registered or do not belong to a Domain, they will be used with the local user security policy and not those of a network user.

### Domain User Password

When entering a User Name and Password that belongs to a domain, the security politics of the Domain network of which the Server is part of will be used.

### User Security

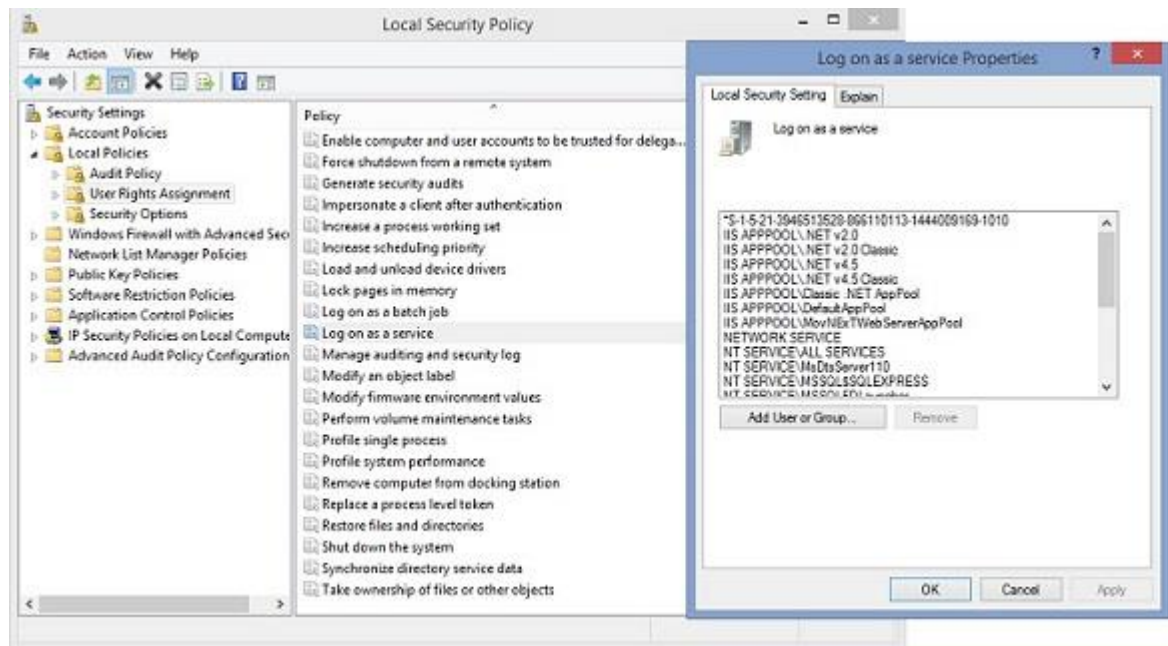
It is important to make sure that the user with which is used to start up the server has been set with the **"Access as Service"** rights. This settings are done in the following way:

1. open the "Control Panel -> Administrative Tools -> Local Security Policy"
2. select "Local Policies"
3. select "User rights Assignment"
4. add user to the "Log on As Service" polity

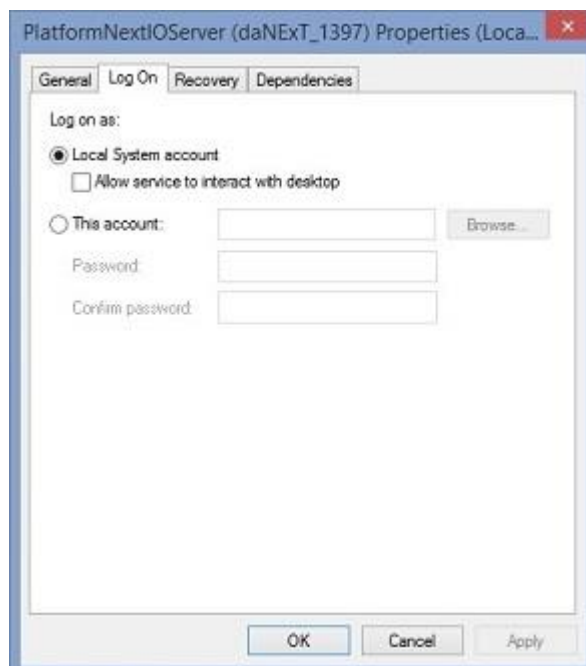
When indicating the User Name and Password of a local user you will need to check whether the user used for installing the service has the necessary requirement to start it up afterwards. These requirements are verified or configured in the PC's local security policy (Security Policy: *Control Panel> Administrative Tools>Local Security Policy>Local Policies>User Right Assignment>Log-On As a Service*)



The Security Policy modification requires administrator user log on.



Once these operations have been completed it will be possible to start the Server up using the "Start Service" command in the window.



*Here the Service properties show that the Server has been installed as Local System Account.*

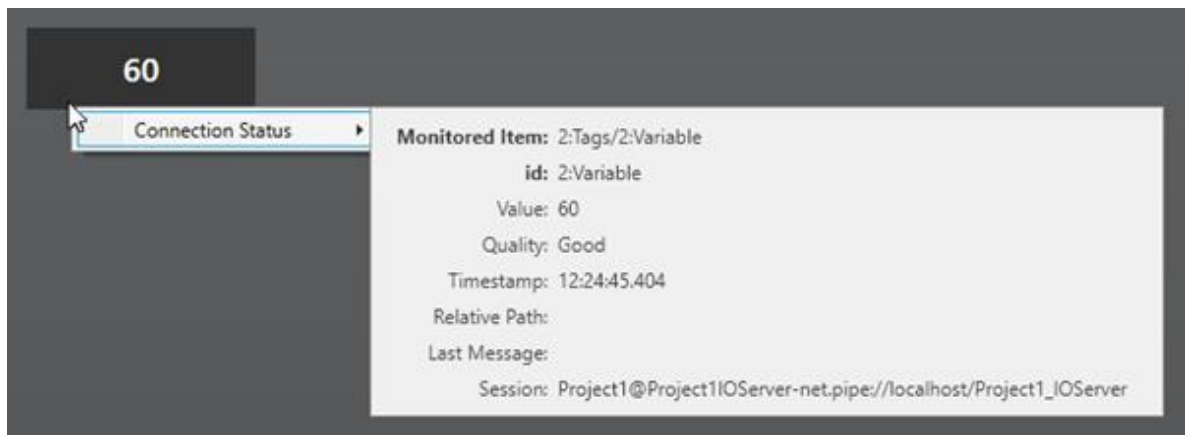
## 17.5. Runtime Diagnostics

### 17.5.1. Connection Status

The Connection Status menu can be activated in Runtime by right clicking on the border of each graphical element on screen and connected to a Tag. This menu can always be accessed when project is started up in design mode or from command line. This menu shows connection information of the tag associated to the object.



The object's border will show red in cases of communication error.



This menu shows the following information:

- **"Monitored Item"**: complete name of variable
- **"id"**: name of variable
- **"Value"**: variable value, if invalid, will be set = null **"Quality"**: communication quality can be Good, Uncertain, BadTimeout, BadOutOfRange...For the complete list please refer to the documentation relating to the OPC UA specifications.
- **"Timestamp"**: date of value update
- **"Relative Path"**: relative path
- **"Last message"**: last message sent by server connection.
- **"Session"**: connection session towards server.

### 17.5.2. Client Status (SHIFT + F1)

The Client Status window can be activated at Runtime with the "SHIFT+F1" command, but it is only accessible when Runtime has been launched from the design environment. The Client Status window will not be available, when the project is started up from command line.

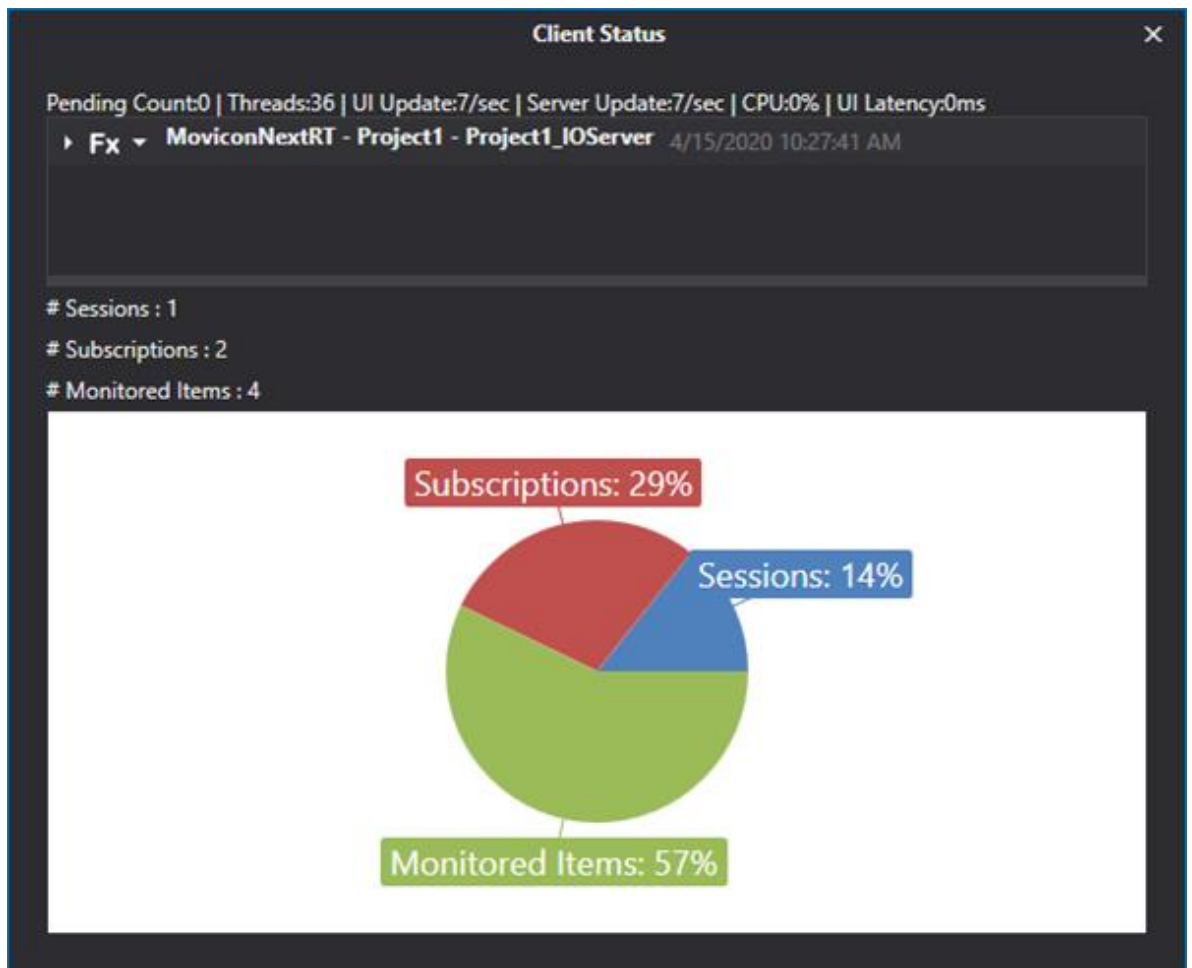
This window reports the open OPC UA sessions between Client and Server with a series of information concerning Threads, CPU occupation percentage and other at the top.

At the bottom, instead, you will find the following information:

- The number of OPC UA communication Sessions that the Client has opened towards the Server
- The number of subscriptions that the Client has activated with the Server. The subscriptions are used to define the data acquisition from the Server.
- The number of Monitored items being how many tags the Client has subscribed to the Server.

The percentage of Monitored Items, Subscriptions and Session are report in the pie chart below:





### 17.5.3. System Log (SHIFT + F2)

The System Log Window can only be accessed in Runtime using the "SHIFT+F2" command when Runtime is launched directly from the development environment. It cannot be accessed when the project is started up from command line.

This window reports all the informative and error messages generated while running the project. There is also a "System Event Viewer" in the toolbox which can be inserted on screen.

System Log								
Path: SysLog.xml		Log Filter: -All-		Best Fit Columns		Select Last Select First Select All		
Item	Logger	TimeStamp	Level	NDC	Identity	Message	Details	MachineName
2587	Generale	5/13/2021 9:01:51...	INFO			Start		ThomasM
2588	License Manager	5/13/2021 9:02:12...	WARN			License Not Found!		ThomasM
2589	License Manager	5/13/2021 11:02:1...	WARN			License Not Found!		ThomasM
2590	Generale	5/17/2021 9:35:44...	INFO			Start		ThomasM
2591	License Manager	5/17/2021 9:36:13...	WARN			License Not Found!		ThomasM
2592	License Manager	5/17/2021 11:36:3...	WARN			License Not Found!		ThomasM
2593	Generale	5/18/2021 9:31:12...	INFO			Start		ThomasM
2594	License Manager	5/18/2021 9:31:40...	WARN			License Not Found!		ThomasM
2595	Generale	5/18/2021 9:53:09...	INFO			Start		ThomasM

The data shown in the "System log" window is saved on file. This file is created in:

C:\ProgramData\Progea\Movicon.NExT.Log\Log.xml.

The information saved in the Log file is displayed as follows:

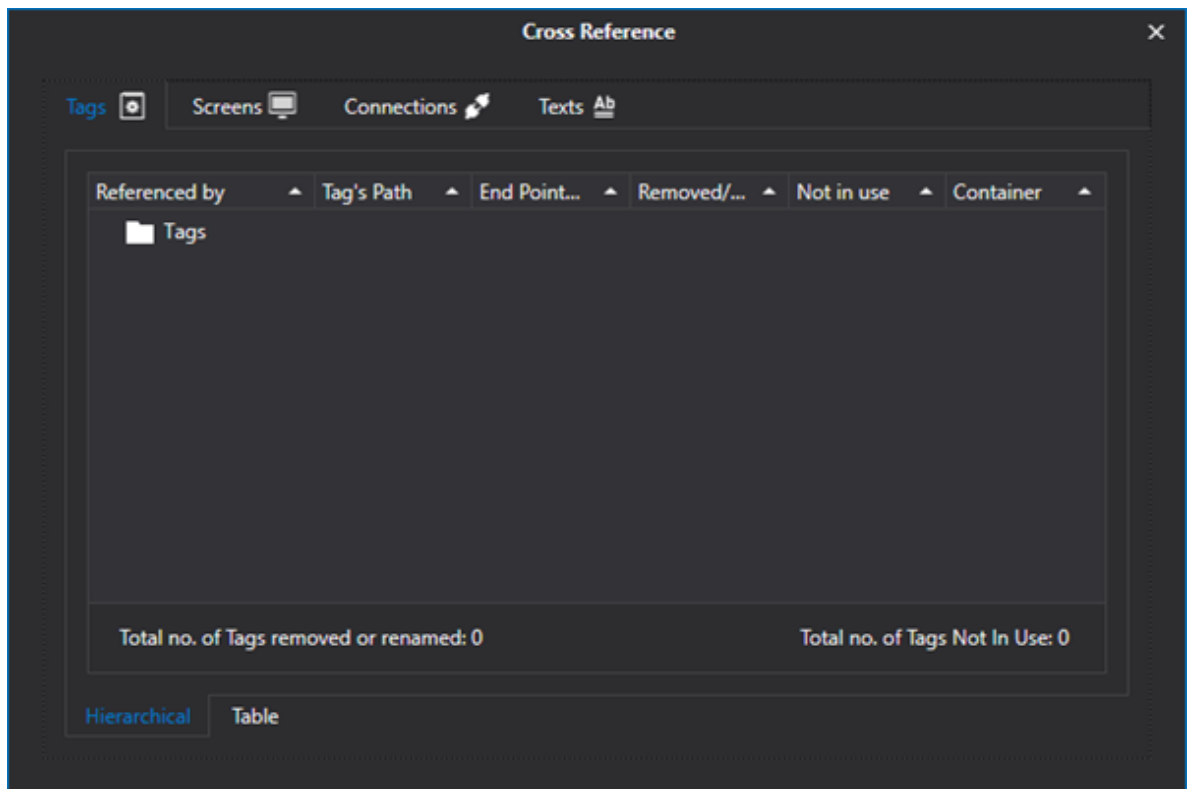
<b>Logger</b>	Module from which message derives, for example Server if from Server side or General if a project message.
<b>Item</b>	Generated message is shown by incremental ordinal number of occurrence.
<b>TimeStamp</b>	Date and time when message was generated.
<b>Level</b>	Indicates the type of message, INFO if it is a simple informational message, ERROR if it is an error
<b>NDC</b>	Work in progress
<b>Identity</b>	identified the user who executed the process that generated the error.
<b>Message</b>	Error message text.
<b>MachineName</b>	Name of PC in which message was collected.
<b>HostName</b>	Name of PC in which message was generated.
<b>UserName</b>	Name of Windows user which Movicon started up with.
<b>App</b>	Application which generated error (MoviconNExT.exe, MovNExTServer.exe...).

In addition to messages the System Log window also has some fields that can be used for filtering events for the "Logger" column or for searching texts.

#### 17.5.4. Cross Reference (SHIFT + F3)

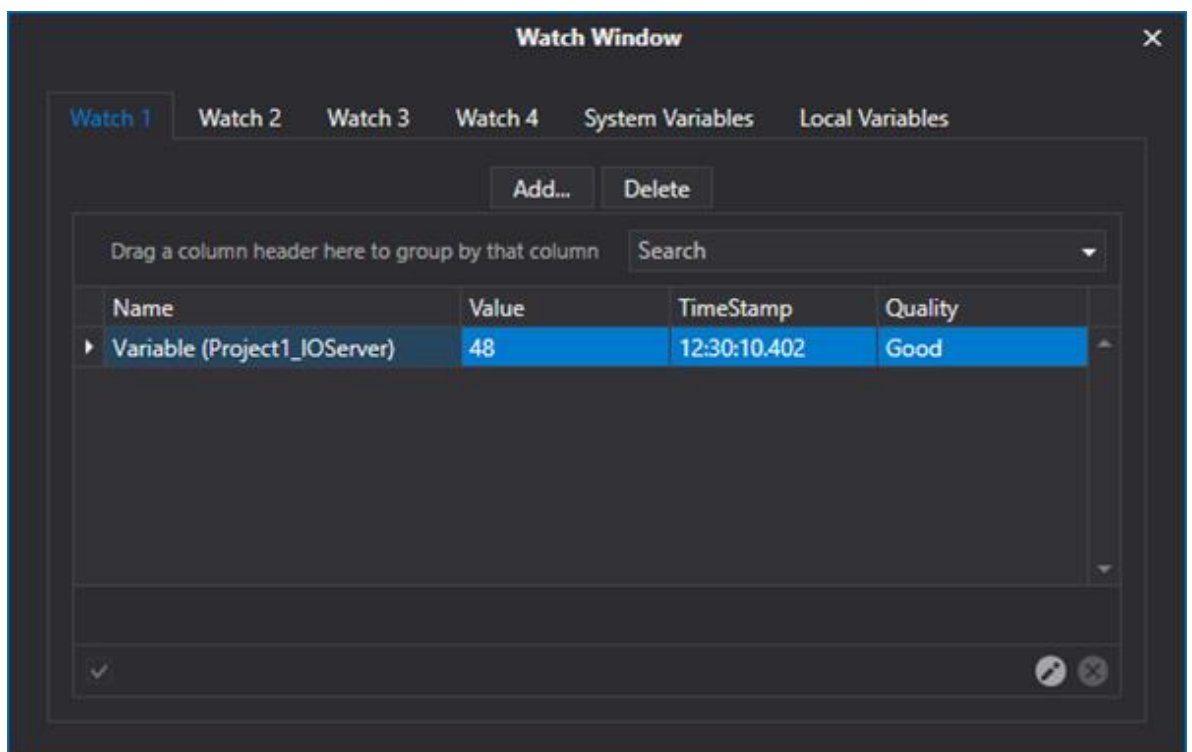
The Cross Reference window can only be activated in Runtime using the "SHIFT+F3" command when Runtime has been launched from the Development environment. The Cross Reference Window is not available when the project is launched from command line.

When the "Update Cross Reference" command is executed first in the development environment, the Cross Reference window will consent the project to be analyzed at Runtime by reporting which objects or resources tags are connected to, in which commands screens are called and other.



### 17.5.5. Watch (SHIFT + F5)

The Watch Window can only be activated in Runtime using the "SHIFT+F5" command when the project has been launched from the development environment. This window cannot be accessed when the project is started up from command line.



The Watch Window is used to monitor and modify project variables whether they be server variables or local client variables. The following tabs will be shown in the window:

- Four Watch tabs (Watch 1, Watch 2 and so forth) from which it will be possible to select the Server variables that have been defined in the project.
- A SystemVariables tab containing a list of the system variables.
- A LocalVariables tab containing a list of the Local Client variables.

When the Watch Window is closed, the settings performed in the Watch tab will be saved in the Isolated Storage of the Windows user. In this way it will be possible to keep the list of tags inserted in the Watch tabs for when the window is opened next time.

#### Limitations

Method variables are not handled in the Watch Window.

## 17.6. PC Time Changes

When the time changes on the PC, the Server will disconnect the tags and the reconnect them. This behavior is desirable but it would be a security concern if the connection was allowed to continue when stepped forward to change dates. Many connected security elements that might go into timeout or expire altogether will have to be closed and recreated.

Even though these situations normally happen only when changing from standard time to daylight saving time (DST) and vice-versa, we recommend that you take precautions to avoid scripts going into error when attempting to access tags the exact moment they are disconnected. This can be done by providing an error management with the script.

# 18. License Manager

## 18.1. Introduction to using Licenses

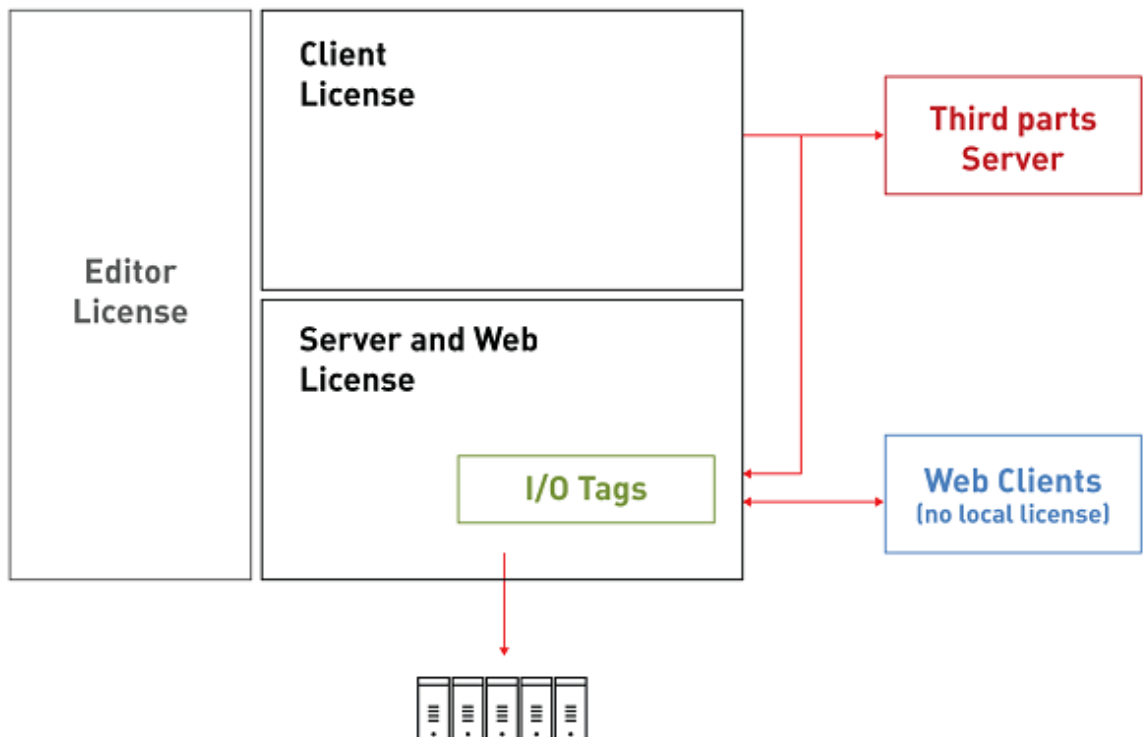
The Movicon.NExT software product is protected against unauthorized use by means of a hardware or software license. The license that authorizes the use of the software product can be purchased in two different modes according to customer needs:

1. **Hardware USB dongle:** The hardware USB dongle is a floating license which is easily transportable from one PC to another PC and is free of any problems that may occur on the machine. However, the USB dongle does carry the risk of being lost, stolen or damaged.
2. **Software Key:** The software key does not carry the risk of being lost, stolen or damaged and even though it can be moved from PC to another PC, it is not as easy to do so compared to using the USB dongle.

At startup and during operation, the system makes a sequence of checks for the presence of a local hardware USB dongle, a local software key and networking license respectively.



If the system does not detect the right license at startup, it will switch to Evaluation Mode with the functions described below.



### Editor License

The Editor License, which comes in the form of both hardware or software, is needed to use the product in the design phase.

In this phase the developer will be able to enter the development environment (Editor) to configure and program the project's functions to be used at runtime.

## Runtime License

The Runtime License, which comes in the form of both hardware and software, is needed to run the project in Runtime by either launching it from the development environment or command line. The Movicon.NExT platform's Runtime Licenses are modular and scalable and are also available as Server only or Client only. They can be purchased based on one's project needs in order to contain costs as much as possible according to the official price list modalities. Project Runtime will always require the presence of a regular Runtime License. Without a Runtime License, the project will run in Demo mode with certain limitations as described below.

## Evaluation Mode

Without a regular Editor Licence or Runtime License both the development and Runtime environments will run in Evaluation Mode (Demo).

The product **can be used in Demo mode only for evaluation purposes and not for professional use according to the terms and conditions stipulated in the user license (EULA).**

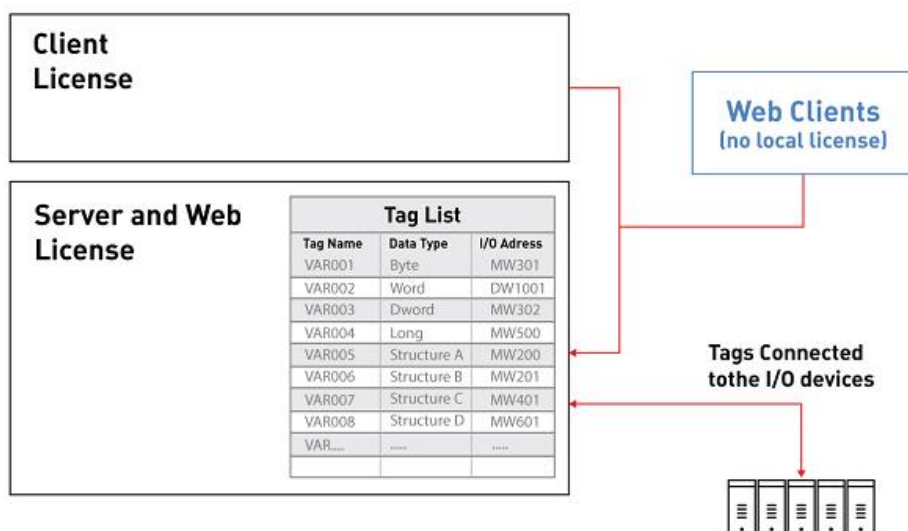
While running in Evaluation Mode:

- **Development environment:** all the features can be used and therefore projects can be created, opened and edited without being restricted on the number of Tags, Screens, Alarms that can be used. A warning will be displayed at increasingly frequent intervals to remind users that they are currently in Demo Mode.
- **Runtime:** The runtime limit is a continuous 2 hours. All the functions are available during this time but the number of variables, screens and alarms is limited.

## 18.2. I/O Tag Count

The Movicon.NExT Runtime Licenses are sized base on the number of project I/O Tags (Variables). All the Tags defined in the Server and connected to the field, being those set with the "Physical I/O Address" property, are counted for License purposes.

At startup a warning message will show if the number of Tags connected to the field exceeds the value defined on the License and the server will not be started up.



## Tag Definition

For License clarity, the Tag definition is used and stands for any variable type: bit, byte, word, double, long. This means that each Tag can define data up to 64 bits.

Each member of Structure or Array variable types is counted as one Tag for license purposes.

**1 Tag = Each individual data type in a variable from 1 to 64 bits.**

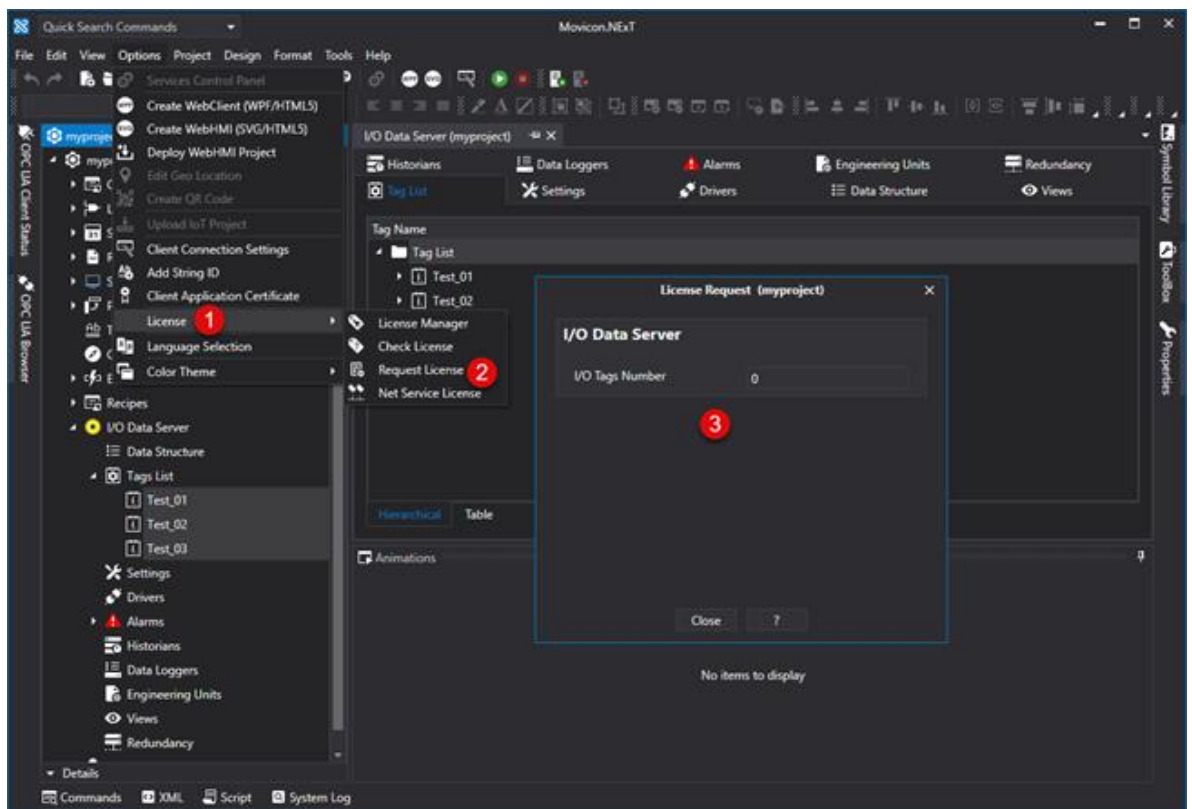
**Note: Each array or structure member is considered as 1 Tag**



Each Tag, independently from the fact that it is byte, word, long, double type, can be pointed to within its single bit in accordance with the appropriate syntax provided for assigning Tags in objects.

## License Tag Count

For license purposes Movicon.NExT provides a tool that allows the developer to know how many Tags are currently defined in the project. This tool is accessed from the menu Options > License > Requested License as shown in the image below:



This window can be only accessed when the project is open and the name of the project is selected in the Project Explorer window.

## 18.3. Installed License verification

By using the "Find License" command from the "Options > License" Menu, you will be able to view all the data relating to the installed license to see if it has been activated correctly.

Whether a hardware or software key, the "Find Licence" window will report the following details indifferently:

- Expiry date (if provided)
- serial number
- all the enables functions

Since the license can be Client+Server type and also "Client Only" or "Server Only" type, the optional functions will vary according to type.



A "License not found" message will display at the top of the window when no license is presence and the "Demo Mode" notice will show at the bottom of the window. All the license options will be enabled in "Demo Mode" and the counters will be set with the maximum value allowed but the system will only function for two hours in Runtime.





When opening this window will check for installed licences and report the relating characteristics according to the functions indicated:

#### Server License Tags Nr.

This field displays the maximum number of Tags that can be used by the Server Module. This count is performed on those Tags defined in the Server's Address Space and connected to the field.

#### I/O Drivers Nr.

This field displays the maximum number of Communication Drivers that can be used by the I/O Data Server in the project.

#### Child Projects Nr.

This parameter indicated the maximum number of child projects that can be declared in the project.

#### HMI Screens Nr.

This field displays the maximum number of Screens that can be used in the Movicon project for the platform's HMI Client. This number can be exceeded in design mode but not in runtime mode. Once this limit is exceeded in Runtime mode, no other screen can be opened.

#### Net Client Nr.

This parameter indicates the number of contemporary users or Windows Terminal Server users per NET license.

### Alarms Nr.

This field displays the maximum number of Alarms that can be used in the project. This number can be exceeded in design mode but not in runtime mode which will ignore and not manage those alarms exceeding the maximum limit set.

### I/O Data Server Instance Nr.

This parameter indicated the number of I/O Data Servers that can be startup at the same time on the same PC.

In cases when the value is zero but the "Runtime IO Data Server" option is active, at least one I/O Data Server will nevertheless be run on the PC.

### Net Client Nr.

This parameter indicates the number of contemporary users or Windows Terminal Server users per NET license.

### Pro.Energy Measures No.

This License parameter indicates the maximum number of measures to be managed in the Pro.Energy project.

### Pro.Lean OEE Nr.

This license parameter indicates the maximum number of resources/machines that are to be managed with the OEE within thePro.Lean project.

### Drivers

The driver licence is management is subdivided into 5 macro classes in this section.



The "Automation" and "Modbus" are always available for the **Movicon NeXT** and Connex versions. While the "Facilities" and "ModBus" options will be activated for the **Movicon NExT BA** version. All the others will need to have the "Runtime License Option" activated according to the driver you wish to use.



In cases when the driver selected does not have the license activated for the group it belongs to, a "**The driver[nome driver]has not been initialized properly...**" message will appear.

### Editor

When enabled this option uses the project in Design Mode to allow the editing and developing of any project type.

### I/O Data Server Runtime

When enabled, this option allows runtime execution of the I/O Data Server module.

Note that the I/O Data Server module must be enabled for the Alarm and Historian Managements.

### Movicon HMI Client Runtime

When this option is enabled, the MoviconNExT module can be executed in runtime, therefore the HMI client towards the PlatformNext server or other servers.

### Historian Manager

When enabled, this option consents runtime execution of the Historian, a function provided by the I/O Data Server module. The Historian provides the use of Data Logger and Historical recording engines. If this option has not been enabled on the license, the server's data recording engines will not startup in runtime.

### **Recipe Manager**

When enabled this option consents execution of the runtime "Recipe" modules. The recipe modules allow the user to manage and view archived data and direct communications with field devices.

### **.NET Scripts**

When enabled this option consents the "VB.NET Script" to be used in runtime as a resource both on server and client. It can also be used as code in screen objects. This means that the Script code interpreter engine is enabled for the entire system.

### **Scheduler**

When enabled this option allows the command scheduler server, i.e. the Scheduler Module, to be started up in runtime. If this option is missing, the Scheduler Module will startup in Demo Mode and work for two hours only.

### **Networking**

When enabled, this option allows the Movicon.NExT HMI Clients to connect to the I/O Data Server in runtime when the Movicon.NExT Clients reside in other workstations on the net.

If this option is not enabled, only local clients in the same PC will be able to connect to the Server.

### **Reports Manager**

This option enables the used of the Data Analysis and Report objects within the project. When is option is disabled the Data Analysis objects will remain disables and the Reports will not be available for use.

Please note that the Reports are available independently from the licence with using Web Client.

### **Alarm Dispatcher**

When enabled, this option allows the Alarm Dispatcher module to startup in runtime to send alarm and message notifications. Without this option enabled, the Alarm Dispatcher Module will startup and function in Demo Mode for two hours only.

### **Downtime Analyzer**

When enabled this option allows statistical project Alarm Reports to be displayed in runtime to analyse system alarms and downtimes according to the analysis graphs and tables provided to show event frequencies and durations based on set time ranges.

### **OPC UA Server**

This option is used to enable the OPC UA Server runtime functionality. In this case the I/O Data Server will expose its Tags to OPC UA Clients (i.e. third party software applications, third party HMI etc.), using the OPC UA connectivity standard specifications.

When this option is disabled the project Tags will not be exposed to other third party OPC UA Servers.

### **Redundancy**

When enabled this option exposes the Server redundancy functionality locally. The redundancy function (server hot backup) manages two or more servers in 'fault tolerant' mode by synchronizing both real-time and historical data.

Please note that this functionality must be enabled in each individual server in order to synchronize. This functionality is continuously active and available in Client stations for dynamic and intelligent data search and exchange with the active server.

### Geo Localization

When enabled this option allows the Geo-Localization feature to function in runtime on project side. The Geo-Localization feature includes the use of geographic maps and their related features.

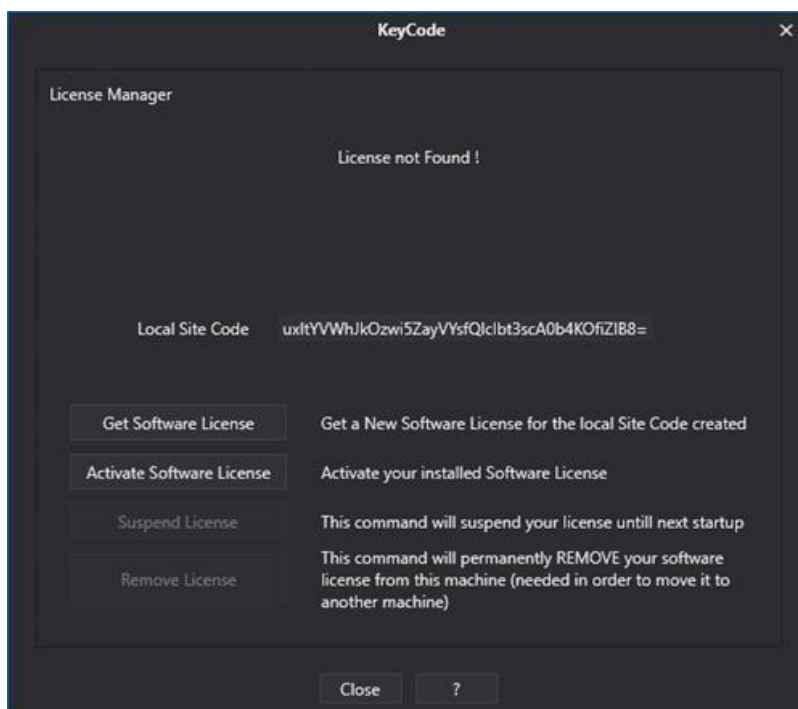
### 3D Manager

When enabled this option allows the Movicon.NExT 3D graphics manager to be used in runtime. If this option is not enabled, it will still be possible to insert 3D objects during design mode but they will not be displayed in Runtime.

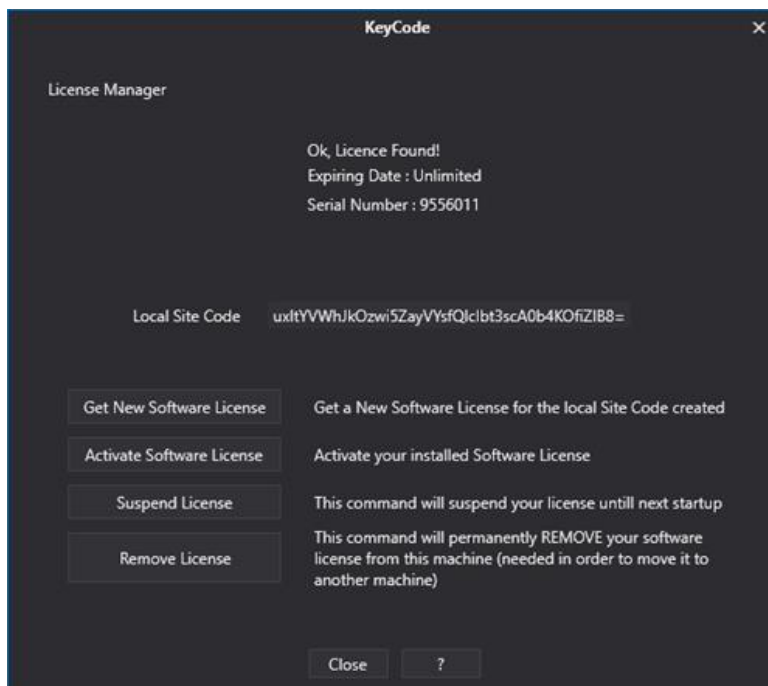
## 18.4. License Manager

Designers can access window used for managing licenses by using the "**License Manager**" command from the "Option > License" menu to activate and manage the software license. The software license is a file containing a unique key code which is determined and generated as a consequence to the local Site Code. This unique key code must be obtained from Sales Department by providing them with your Local Site Code which is available from the License Manager window.

If no license can be found a "License not Found !" message will show and the product will run in Evaluation Mode for the purpose of testing and evaluation according to the terms and conditioned stipulated in the user license. You can also use the Manger License window to request or activate a software license.



When a valid license is found, the system will report all of its information with the option to suspend or remove it.



The License Manager commands are described below:

### Expiry Date

It is possible to have a temporary license that is valid for a limited period of time. In this case, based on the sales agreement, the user can install a license that will automatically expire when the fixed time is reached. Information relating to the activated temporary license's expiry can be seen in the License Manager window.

### Serial Number

The Serial Number shown in the License Manager window is used to identify the license. This serial number will be required by the Progea sales offices or distributors when license needs to be modified or removed.

### The Site Code

A Local Site Code alphanumeric string is displayed in the License Manager window. This code is essential for obtaining a valid activation license. The Local Site Code is unique and tied to the machine in which the product is installed.



The software license is unique and exclusively tied your 'Site Code' which is the local code belonging to the PC from which it was generated. Therefore this code will not work in any other PC.

In order to get a software license you must purchase one from the Sales offices or from one of the network of distributors. The user who purchases a software license will receive a license serial number and a file containing the activation code generated in function with the Site Code. Otherwise they will receive instructions to access the website in order to generate the license online.

### Obtaining a new software license

The **"Get New Software License"** command in the License Manager window gets you access to the website reserved for registered Users in order to generate product licenses online. Once logged on the

authorized user can digit the purchased license serial number as instructed by the guided procedure in order to immediately generate the file with the license code to insert and activate in their system.



The software license therefore physically exists in an encrypted file containing the purchased license code which is determined by the local Site Code.  
The License file can be generated autonomously on the website or received by email directly from the sales offices.

A backup copy of the license file can be made as pleased but the license cannot be duplicated in other PC as it is tied to the local site code of the PC where the software is installed.

### Activating the software license

Once you have received the software license file, you will need to activate the license.

This is done by using the "**Activate Software License**", command from the License Manager window. You will be asked to select the license file that you have received. Therefore you must indicate the destination folder of this file which usually has the .mlztrack extension.

Once confirming the selection of this file, the system will activate and register the license by checking it with the Local Site Code.

If this procedure ends successfully, the 'License activation was successful!' message will appear and the product will be correctly licensed for use.

Once this has happened you will be able to view all the details of the installed license in the "Find License" window as described in the below paragraphs.



The 'Activate Software License' must also be used when upgrading the license, for example after purchasing new functionalities or increasing the number of tags.

### Suspend License

This command can be used for deactivating the installed license until the next startup. Once you have suspended the license it will function temporarily in "Demo Mode". This command can be used at your discretion for any debug or maintenance activities.

The installed license will be restored and activated when restarting Movicon.NExT again.

The license can also be suspended for the a project's runtime session.

This is done by simply keeping the "D" key pressed during the project startup phase in runtime. The project will then startup in Demo mode.

To restore the license, simple restart the project in runtime as is normally done.

### Remove License

The 'Remove License' command allows you to remove and invalidate the installed license in order to move it to another PC.

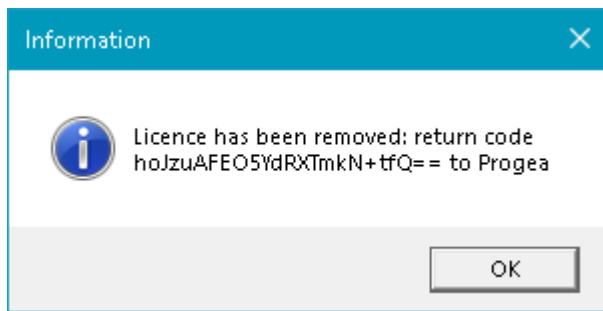
When removing license to upgrade it, use the software license activation procedures.



**IMPORTANT:** After license removal has been confirmed, a dialog window will display showing the return code to use to obtain a new license file containing a new Serial Number and Site Code to use in the new PC.

Once the removal procedure has been completed, the system will return back to 'Demo Mode' and it will not be possible to reuse the previous license file again.

However, a different software license can be activated as long as it refers to a different Serial Number than the one used before.



### Temporary License

You can obtain a temporary license which is used with a set time limit. In this case, in accordance with the sales agreement, you will be able to install and use this kind of license until its preset time limit expires. The temporary license expiry date will be shown in the License Manager window.

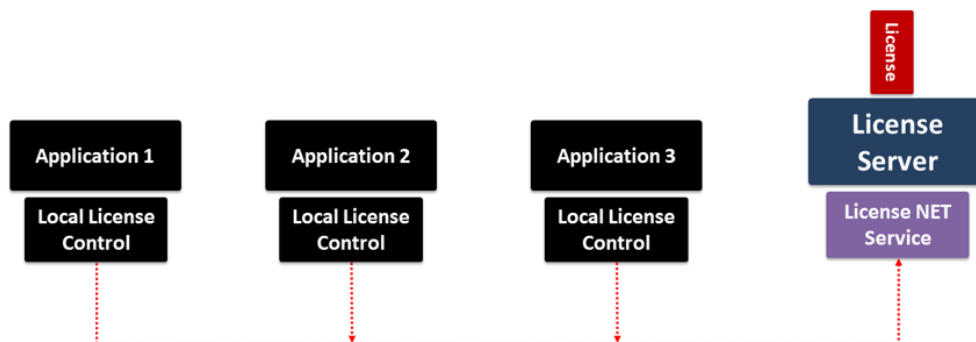
### License file location

Once the software license is activated, the license file named "msz\_controls.xml" is saved in the folder: %ProgramData%\Progea\Movicon.NExT\ . If this file is removed or moved, the software will work in EvaluationMode.

## 18.5. NET Licenses

The Movicon.NExT license manager allows net licenses to be managed in a manner that enables better support to those architectures that provide the use of a certain number of network stations with the net license centralized **on one Server station**, or those architectures based on **Windows Terminal Server** (or Remote Desktop).

Architectures such as these require the use of the Movicon.NExT NET license services.



*This diagram shows an example of stations distributed in a network where each application searches for an active net license.*

It is therefore possible For those who wish to use a one only centralized license in the network architecture instead of one individual license for each station. In this way each remote user can run the software by logging on with one unique license on the net.



Licence sharing using the NET License Server is available for both the Hardware USB dongle and the software key.

In any case, an adequate number of "NET License Users" will need to be activated on the license.

### How the NET License works

The Movicon.NExT Licence service creates a list of connected Clients and once the limit set on the license has been reached, it will not accept any other connections: in this case, the application not authenticated because the set number of Clients has been reached will run in Demo mode due to not receiving license activation.

The NET License Service manages connected and authenticated Clients by adding clients when connecting and removing them when they disconnect (e.g. when a Movicon.NExT application closes correctly in net, the Licence network Service updates the list to allow other Clients to connect).



In this way, the network manager offers dynamic license use so that systems can commit or free the license as needed.

However, it is be to keep in mind that opening more than one product instance on the same PC (using the same user), each instance will occupy one licensed Client.



Attention, certain circumstances may arise where a client does not notify its closure correctly, for example when an application is aborted manually or is closed in error. For this reason, once a Client requests access to the network service, it will check to see if the Client is already on the list and if so it will only update the Client's date of access.

## NET License Service Configuration on Server

To made the net license available on all remote stations, you will need to first install the net licence service on the PC Server by opening the "NET License Service" window from the "Options > License" menu.

The image shows a screenshot of the "NET License Service" configuration window. The window has a dark gray background and a title bar with the text "NET License Service". It contains several input fields and buttons. At the top, there is a label "Enter credentials for a domain user (domain\user or user@domain)". Below this are two input fields: "User Name:" and "Password:". The "Password:" field is masked with dots. Below these fields are four buttons: "Install" (with a download icon), "Uninstall" (with a red X icon), "Start" (with a play icon), and "Stop" (with a stop icon). Below these buttons is another label "Enter settings for NET dongle connection". Below this label are two input fields: "Server Name:" and "Port Number:". The "Server Name:" field contains the text "localhost" and the "Port Number:" field contains the text "62873". At the bottom of the window are two buttons: "Save Settings" and "Reload Settings".

By means of using this window you can execute the commands relating to the license service on the station you are operating from:



- **Install:** Installs the NET License application as a local Windows service. The service is installed to be run as "Local System" user, but it is also possible to install the service in a way so that is used another user (local or domain) by means of using the Username and Password fields.
- **Uninstall:** Removes the net license service.
- **Start:** Starts the service application for the NET License.
- **Stop:** Stops the service application for the NET License and the parameters can be configured.
- **Server Name:** defines the PC Server's name on which to run the NET License Service and from which licenses are shared. It is advised to leave the "localhost" value set for default on the PC Server.
- **Port Number:** defines the port to be listened to by the NET License Service (default 62873)



On the computer where the licensing service is installed, it is a good idea to create a rule on the firewall to allow clients access to the licensing service. If the proposed default port 62873 is kept, you will then need to create a rule to accept incoming TCP and UDP connections on this port.

## Client Configurations

To use a NET License from any one of the Client stations simply:

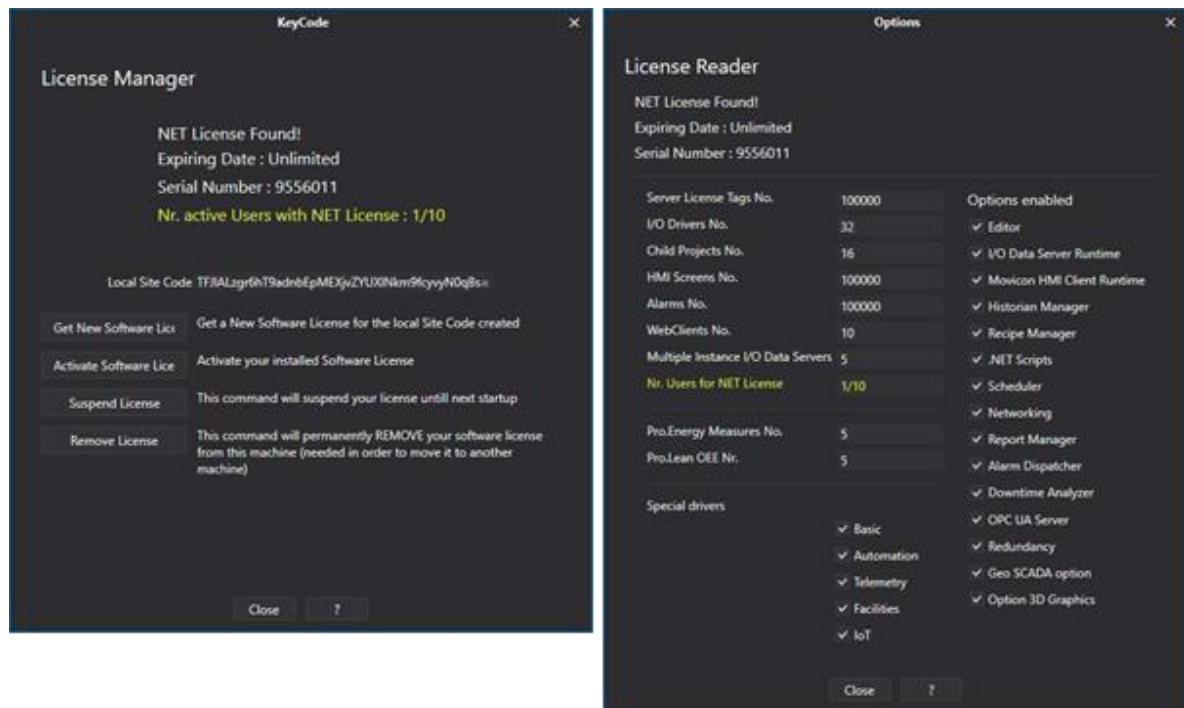
1. Open the "NET License Service" window from the "Options > License" menu
2. In the Server Name parameter, enter the name of the PC where the license is installed and where the NET License Server is installed
3. Indicate the port on which the NET License Service will listen (default 62873).
4. Confirm by using the Save Settings button.

## NET License status info

You will find information of the status relating to the use of the NET License in the "License Manager" and "License Reader" windows of each station.

In these windows you will be able to see:

- **No. of Users currently active with the NET Service**
- **Maximum No. of users provided with the purchased and installed NET License.**



Keep in mind that the License Manager window will always show the "Local Site Code" as well. Each Site Code is local regardless of the net license. In fact a Local station can always install local license if needed regardless whether or not a net license has been installed.

## Installing a NET License Service manually

Up to now we have described how to install the NET License Service and the way it functions using the Movicon.NEXt product and its tools. Therefore, the end result would be to have the Movicon.NEXt installed in the Server station hosting the Net License.

However, the Net license service can also be used without having to install the Movicon.NEXt locally. This can be done by an **expert user** by taking into account the above and the following:

1. The MSZService.exe application and its MSZService.exe.config file can be copied and installed manually in the PC station that manages the net license. This executable can be started up as a Windows Service or from console (arg: -noservice). In the second case the local SiteCode and other useful information will be displayed in the DOS window. This information will include the number of enabled users (when license is acknowledged) and the number of users who have used the service.
2. This local Site Code can be used for obtaining a Net License and which must be copied in the C:\ProgramData\Progea\Movicon.NEXt folder manually.
3. Expert users can change the port (ServerPort) settings, which will be used by the service to accept User authentications, directly within the MSZService.exe.config file.

## NET License for Windows Terminal Server or Remote Desktop systems

Using architecture based on Microsoft Terminal Server (WTS) or Remote Desktop Server operating systems requires a NET license as described above.

The number of NET license users will correspond to the number of Terminal Server Users that the Server architecture has to manage.

## 18.6. WebClient Licenses

When a Movicon.NExT project needs to have WebClient access only (either by HTML5 or App), and therefore without any desktop client connections, the license can be configured by inserting only those options needed for WebClient. In this case, for example, you will not need the "Runtime HMI Client" license option, but you will need to define the No. of WebClient Users to connect simultaneously and the No. of Client License Tags to connect at each session. You will also need to enable other additional options such as the "Report Manager" if you wish to see historical data using the Data Analysis or Reports for example.

### Configuration with WebServer and I/O Data Server in separate machines

When the system has been configured to startup the project's I/O Data Server in one machine and the IIS WebServer or the WebClient(WPF) App Service in another, you will also need to insert a license in the machine where the IIS WebServer or the App Service is being run. To enable this the license should be configured with the features needed for WebClient, as described above.

In a configuration like this one the following will need to be inserted:

- a 'Server' license in the machine where the I/O Data Server is running without specifying the 'No. of Client License Tags' and 'Tag No. of WebClient Users'.
- a license in the machine with the WebServer (IIS) or App Service configured with only the 'No. of Client License Tags' and 'Tag No. of WebClient Users' options for WebClient, plus any other options needed such as the "Report Manager".

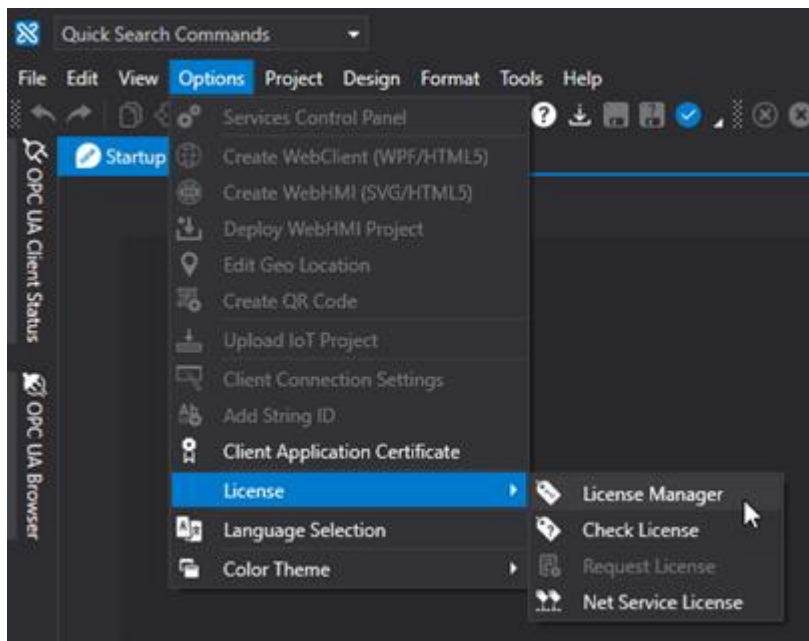
As an alternative, you can also use a centralized license by purchasing a "NET" license to install in any one of the machines and to which all the I/O Data Server and WebServer instances, for example, will connect. In this case the license could also include all the options needed.

## 18.7. License Activation

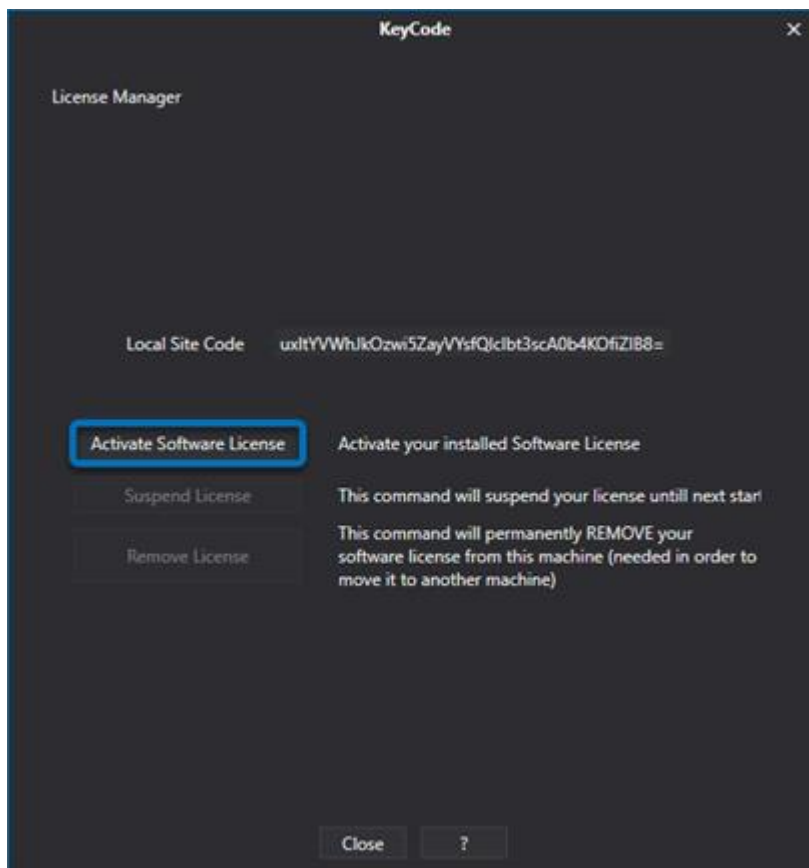
### How to activate license on local Windows PC

Once Movicon.NExT (or ConnexT) has been installed on the PC where it will be launched into, open the Movicon.NExT editor:

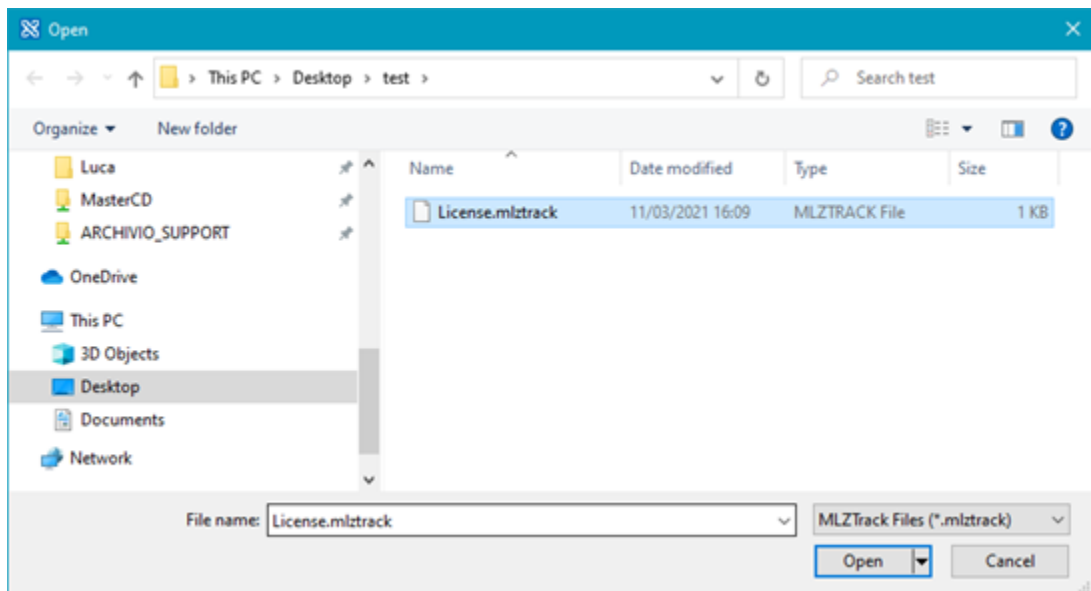
1. Go to menu and select: "Options > License > License Manager"



2. Select "Activate Software License" in the License Manager window.



3. Another window will open allowing you to select the license file with the "mlztrack" extension:



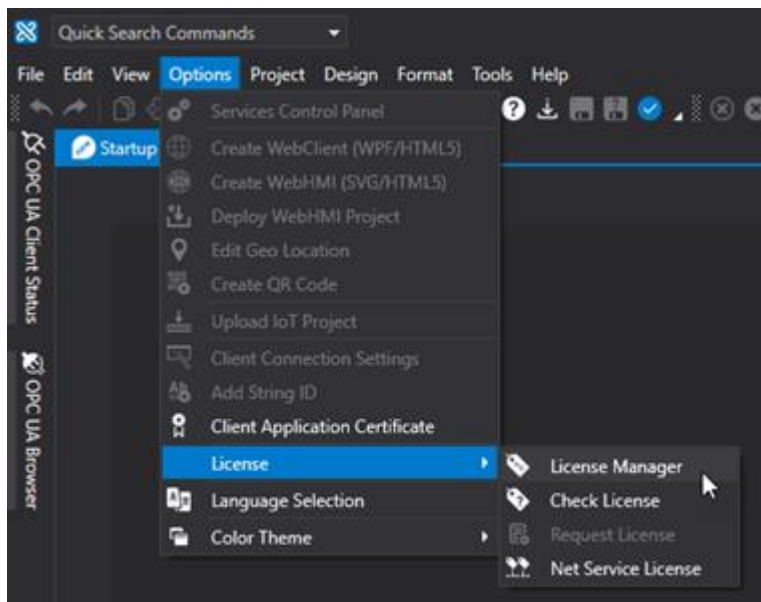
Confirmation will be given once the license has been installed.

You will be able to check the installed license contents by using the "Options > License > Check Manager" menu.

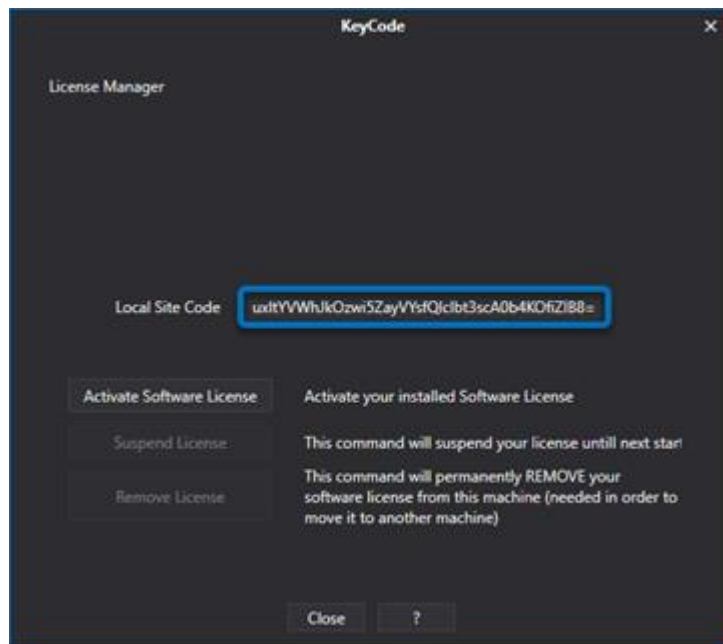
### How to retrieve the Site Code on local Windows PC

Once Movicon.NEXT (or Connexx) has been installed on the PC where Runtime will be launched, open the Movicon.NEXT editor:

1. Go to menu and select: "Options > License > License Manager"



2. You can read and copy the "Local Site Code" string from within the License Manager window to use for generating the license.



# 19. Movicon NExT Builder

## 19.1. Introduction

The Movicon.NExT projects can also be created using automatic procedures that can be completely customized. For instance, wizards can be autonomously created to create or edit Movicon.NExT projects and components.

This offers the expert user a great way to develop projects or repetitive project parts much faster by creating custom wizards that will reduce development times to the minimum.

For instance, projects can be automatically created with screens, variables, alarms, Historian and many other project functions by simply creating the rules to automatically create the automatic creation procedures.

This powerful tool is based on a component designed for this purpose and which can be managed with Movicon VB.NET script or by autonomously creating the tool externally using Visual Studio. This tool is called the Movicon.NExT Project Builder. For example, some users use it to create automated procedures for creating projects or parts of projects, simply by selecting the import sources (e.g. Excel files or external databases). The less expert user may prefer to use the configuration Wizard to define the things they wish to create for the Project Builder to then create the project and all the resources already appropriately configured as required.

### The Project Builder

The Project Builder tool is actually an object (.DLL) that is called when needed by using code. It can be used to create or modify the various Movicon.NExT project resources to be executed, by using the actual project's VB.NET script code, or by autonomously creating an executable tool with Visual Studio. In both cases, the MoviconNextBuilder.dll component may be managed with its own code to automatically create or edit a Movicon.NExT project, or setup guided procedures (wizard) to help users to create and configure the project or part of it.

The Project Builder tool offers the following functions:

#### I/O Data Server

All properties belonging to Tags and Alarms can be edited/customized.

- Create/edit/delete a tag
- Associate/remove an alarm to/from a tag
- Associate/remove an historian to/from a tag
- Create/remove/delete a folder from the tag list
- Create/edit/delete a datalogger
- Add/remove column to/from a datalogger
- Create/remove/delete an alarm area
- Create/remove/delete an alarm source

All those properties belong to alarms can be edited/customized.

- Create/edit/delete an alarm
- Create/edit/delete an Historian
- Create/edit/delete an Engineering Unit
- Edit the IO server settings (Application Name, etc)
- Add/configure/remove a communication driver

#### Screens

- Create/edit/delete a screen
- Insert/edit/delete a toolbox object from the screen
- Insert/edit/delete a library symbol from the screen

### Texts

- Add/edit/remove a string
- Add/remove a language

### Parameters

- Create/delete parameterization file
- Create/delete a parameter (copy alias/variable) from a parameterization file

## 19.2. Visual Studio Project Settings

The Visual Studio project must be set for Net framework 4.6.1 or later versions using the Movicon.NExT installation folder (C:\Program Files\Progea\Movicon.NExT) as the output path.



In order to save the destination folder you may have to open VisualStudio as pc administrator depending on the path in which Movicon NExT was previously installed.

You should find the **using System.Reflection** and **using CommandManager** declarations along with the following **References** in the Visual Studio project:

- **CommandManager.dll** found in the default path: C:\Program Files\Progea\Movicon.NExT\Managers
- **DataLoggerModel.dll** found in the default path: C:\Program Files\Progea\Movicon.NExT
- **DevExpress.Xpo.Vx.x.dll** found in the default path: %windir%\Microsoft.NET\assembly\GAC\_MSIL
- **MoviconNextBuilder.dll** found in the default path: C:\Program Files\Progea\Movicon.NExT
- **OPCUAViewModel.dll** found in the default path: C:\Program Files\Progea\Movicon.NExT
- **ScreenParameterSettings.dll** found in the default path: C:\Program Files\Progea\Movicon.NExT
- **ScreenSettings.dll** found in the default path: C:\Program Files\Progea\Movicon.NExT
- **StringModel.dll** found in the default path: C:\Program Files\Progea\Movicon.NExT
- **UFIInterfaces.dll** found in the default path: C:\Program Files\Progea\Movicon.NExT
- **UFUAModel.dll** found in the default path: C:\Program Files\Progea\Movicon.NExT
- **ViewModelBase.dll** found in the default path: C:\Program Files\Progea\Movicon.NExT
- **WPFUtilities.dll** found in the default path: C:\Program Files\Progea\Movicon.NExT

Finally, the following node should be added within the **App.config** file:

```
<runtime>
  <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
    <probing
      privatePath="DocumentManagers;DocumentManagers\LogicExtensions;Managers;Drivers;
      Toolbox;DataSinks;CommonPlugins;DesignPlugins" />
    </assemblyBinding>
  </runtime>
```



## 19.3. Project Instance Manager

In order to operate in a Movicon.NeXT project using the class library described below in this document, you will first need to get the ProjectBuilder object's instance from the MoviconNextBuilder class. In order to do this you will need to insert a specific instance during the VisualStudio application initialization phase as shown in this example:

```
MoviconNextBuilder.ProjectBuilder Builder;  
Uri fileprj;  
try {  
    Builder = new MoviconNExTBuilder.ProjectBuilder();  
    // get the Movicon NeXT project path  
    fileprj = "ProjectPath\\ProjectName.UFProject"  
    // initialize the ProjectBuilder instance  
    Builder.Init(fileprj);  
} catch (Exception eX) {  
    MessageBox.Show("Error in initializing MoviconNextBuilder dll: " + eX.Message, "MovNextBuilderEx",  
    MessageBoxButtons.OK, MessageBoxIcon.Error);  
}
```

At this point you will be able to obtain the Movicon NeXT project manager objects. The objects to use are:

- **IODataServer**: instance of the class with same name that manages all the data server aspects.
- **ParametersContainer**: instance of the Parameters class for screen parameterization.
- **ScreenContainer**: instance of the class with the same name, for managing screens in create and edit modes.
- **StringsResource**: instance of the TextResource class for managing texts and languages.

The objects can be accessed as properties of the ProjectBuilder object.

## 19.4. TextResource

### 19.4.1. AddLanguage, TextResource Method

**Syntax**            System.Globalization.CultureInfo AddLanguage(string language)

**Description**       Adds the indicated language in the project.

Parameter	Description
string language	name of the language/culture to add

**Result** ScreenParametersSettings.Documents.ScreenParametersDocument (instance of an object that represents the created parameter file. Returns null if in error.

#### Example:

```
// defining a Project Builder string resource  
var texts = Builder.StringsResource;
```

```
// add a new en-US language definition
texts.AddLanguage("en-US");
// add a new it-IT language definition
texts.AddLanguage("it-IT");
// saving text configuration change
texts.Save();
```

### 19.4.2. AddString, TextResource Method

**Syntax**      System.Collections.Generic.List<StringModel.UFStringLocaleText>  
                 AddString(string stringid)

**Description**    Adds with indicated id.

Parameter	Description
string stringid	string id to be added

**Result** System.Collections.Generic.List<StringModel.UFStringLocaleText>: List of objects that represent the string ids just inserted. Returns null when in error.

#### Example:

```
// defining a Project Builder string resource
var texts = Builder.StringsResource;
// add a new en-US language definition
texts.AddLanguage("en-US");
// add a new it-IT language definition
texts.AddLanguage("it-IT");
// add a new stringId and Locale string definition for each language
var str = texts.AddString("first_Id");
if (str.Count == 2) {
    str[0].Locale = "Open";
    str[1].Locale = "Apri";
}
// add a new stringId and Locale string definition for each language
str = texts.AddString("second_Id");
if (str.Count == 2) {
    str[0].Locale = "Close";
    str[1].Locale = "Chiudi";
}
// add a new stringId and Locale string definition for each language
str = texts.AddString("third_Id");
if (str.Count == 2) {
    str[0].Locale = "Up";
    str[1].Locale = "Su";
}
// saving text configuration change
texts.Save();
```

### 19.4.3. DeleteLanguage, TextResource Method

**Syntax**                      `bool DeleteString(string stringId)`

**Description**                Deletes the indicated string from the project.

Parameter	Description
string stringId	Name of the string id to be deleted.

**Result**    Boolean

#### Example:

```
// defining a Project Builder string resource
var texts = Builder.StringsResource;
// delete the en-US language definition
texts.DeleteString("second_Id");
// saving text configuration change
texts.Save();
```

### 19.4.4. DeleteString, TextResource Method

**Syntax**                      `bool DeleteLanguage(string language)`

**Description**                Deletes the indicated language from the project.

Parameter	Description
string language	Name of the language/culture to be deleted.

**Result**    Boolean

#### Example:

```
// defining a Project Builder string resource
var texts = Builder.StringsResource;
// delete the en-US language definition
texts.DeleteLanguage("en-US");
// saving text configuration change
texts.Save();
```

#### 19.4.5. GetString, TextResources Method

**Syntax**      System.Collections.Generic.List<StringModel.UFStringLocaleText>  
GetString(string stringid)

**Description** Retrieves the string indicated.

Parameter	Description
string stringId	name of id string to retrieve.

**Result**      System.Collections.Generic.List<StringModel.UFStringLocaleText>: List of objects that represent the strings of the id just inserted. When in error returns null.

#### Example:

```
// defining a Project Builder string resource
var texts = Builder.StringsResource;
// retrieve the en-US locale string
var retStrings = texts.GetString("first_Id");
if (retStrings != null) {
    foreach(StringModel.UFStringLocaleText rs in retStrings) {
        if (rs.Culture = "en-US"){
            MessageBox.Show("String from " + "'first_Id'" + " in " + rs.Culture + " is: " + rs.Locale);
        }
    }
}
```

#### 19.4.6. Save, TextResource Method

**Syntax**      Void Save()

**Description**      Saves all the changes pending in the screen container

Parameter	Description
-	-

**Result**      -

#### Example:

### 19.4.7. SetString, TextResource Method

**Syntax**      `bool SetString(string stringid, string language, string newvalue)`

<b>Description</b>	Changes the indicated string.
--------------------	-------------------------------

Parameter	Description
string language	Language which string is changed to.
string stringId	Name of the string id to be changed.
string newvalue	new string value

Result	Boolean
True	True
False	False
1	True
0	False
any()	True
all()	False

**Example:**

```
// defining a Project Builder string resource
var texts = Builder.StringsResource;
// delete the en-US language definition
texts.SetString("second_Id","en-US","New string value");
// saving text configuration change
texts.Save();
```

### 19.4.8. TextResource, TextResource Method

**Syntax** `TextResource(DocumentManager.ComponentService.IDocument parent, StringEditorManagerComponent editormanager)`

**Description** This is the class builder. An object instance of the class that is already built and created with the initialization procedure described in initialization phase (see `MoviconNextBuilder.ProjectBuilder` project instance builder)

Parameter	Description
-	-

## Result

**Example:**

## 19.5. ReportContainer

### 19.5.1. GetParameterValue, ReportContainer Method

**Syntax**      object GetParameterValue(string reportName, string paramName, [string subfolder = null])

**Description**   Gets the value of the parameter indicated by the indicated report, possibly located within the subfolder.

Parameter	Description
String ReportName	name of the report, must be present in the project
String ParamName	name of the parameter, which must exist in the report
String Subfolder	eventual name of the folder in which the report is located

**Result**      Object: value of the parameter, which must be properly cast to the desired type.

#### Example:

```
// instance of ScreenContainer object
var rptCollection = Builder.ReportContainer;
// adding screen folder
string paramValue = rptCollection.GetParameterValue("ReportName","StringParamName");
```

### 19.5.2. GetReport, ReportContainer Method

**Syntax**      ReportSettings.Documents.ReportDocument GetReport(string reportName, [string subfolder = null])

**Description**   Gets an instance of the indicated report, possibly located within the subfolder.

Parameter	Description
String ReportName	name of the report, must be present in the project
String Subfolder	eventual name of the folder in which the report is located

**Result**      ReportSettings.Documents.ReportDocument - instance of the report.

#### Example:

```
// instance of ScreenContainer object
var rptCollection = Builder.ReportContainer;
// adding screen folder
ReportSettings.Documents.Reportdocument rptInstance = rptCollection.GetReport("ReportName");
```

### 19.5.3. ReportContainer, ReportContainer Method

**Syntax** ReportContainer(DocumentManager.ComponentService.IDocument parent, UFProjectManagerComponent manager)

**Description** It is the constructor of the class. An instance object of the class, therefore already built, is created with the initialization procedure described in the initialization phase (see MoviconNextBuilder.ProjectBuilder project manager instance)

Parameter	Description
-	-

**Result** -

### 19.5.4. Save, ReportContainer Method

**Syntax** void Save()

**Description** Save all pending changes to the ReportContainer.

Parameter	Description
-	-

**Result** -

### 19.5.5. SetParameterValue, ReportContainer Method

**Syntax** MoviconNextBuilder.ReportReturnCode SetParameterValue(string reportName, string paramName, object value, [string subfolder = null])

**Description** Set the value of the parameter indicated in the indicated report, possibly located within the subfolder.

Parameter	Description
String ReportName	name of the report, must be present in the project
String paramName	name of the parameter, which must exist in the report

Object value	value of the parameter to be modified
String Subfolder	eventual name of the folder in which the report is located

- Result**      Object: outcome of the operation. It can take the following values:
- Ok, parameter changed without errors
  - ReportInvalid, The report contains invalid data
  - DataConversionFail, Error converting value to parameter type.
  - ParamSetFail, Failed to modify the parameter
  - NoReport, The specified report does not exist
  - NoData, The report contains no data.
  - NoParameter, the report contains no parameters.

**Example:**

```
// instance of SrenContainer object
var rptCollection = Builder.ReportContainer;
// adding screen folder
MoviconNextBuilder.ReportReturnCode retCode =
rptCollection.SetParameterValue("ReportName","StringParamName","StringValue");
rptCollection.Save();
```

## 19.6. ScreenContainer

### 19.6.1. AddFolder, ScreenContainer Method

**Syntax**      bool AddFolder(string folder)

**Description**      Adds a folder by the name indicated in the screen tree.

Parameter	Description
string name	name of screen folder to be added. Diverse folder levels can be inserted simply by indicating them in the parameter eg: "FirstFolder\\LastFolder" adds the LastFolder folder content in the FirstFolder folder

**Result**      Bool: operation result.

**Example:**

```
// instance of SrenContainer object
var screenCollection = Builder.ScreenContainer;
// adding screen folder
screenCollection.AddFolder("NewFolder");
```



```
// saving ScreenContainer change
screenCollection.Save();
```

### 19.6.2. AddGeoData, ScreenContainer Method

**Syntax**      `bool AddGeoData(string name, double latitude, double longitude, [OPCUAViewModel.OPCUAEntityReference longitudeTag = null], [OPCUAViewModel.OPCUAEntityReference latitudeTag = null], [string subfolder = null], [bool overwrite = False])`

**Description**      Adds a pair of geographical coordinates to the indicated screen.

Parameter	Description
string name	name of the screen in which to insert the references to the coordinates
double latitude	latitude coordinate
double longitude	longitude coordinate
[OPCUAViewModel.OPCUAEntityReference longitudeTag = null]	instance of an object that represents the tag from which to retrieve the value of the latitude, optional.
[OPCUAViewModel.OPCUAEntityReference latitudeTag = null]	instance of an object that represents the tag from which to retrieve the value of the longitude, optional.
[string subfolder = null]	folder in which the screen is inserted in the screen tree, optional.
[bool overwrite = False]	overwrite flag, optional

**Result**      Bool: operation result.

#### Example:

```
// instance of ScreenContainer object
var screenCollection = Builder.ScreensContainer;
// adding geo coordinates
screenCollection.AddGeoData("NewScreen", 12.1, -35.5);
// saving ScreenContainer change
screenCollection.Save();
```

### 19.6.3. AddScreen, ScreenContainer Method

**Syntax**      `ScreenSettings.ScreenDocument AddScreen(string name, string modelname, [string subfolder = null])`

**Description** Adds a screen with the name indicated based on the template modelname.

Parameter	Description
string name	name of screen to be added
string modelname	name of template to be used for the created screen. The template names are the names of the files contained in the folder (and subfolders) ProgramData\Progea\Movicon.NExT.3.0\NewScreenTypes\
string subfolder = null	name of the folder in which the function's screen object is contained.

**Result** ScreenSettings.ScreenDocument: instance of an object representing the created screen. Returns null if in error.

**Example:**

```
// instance of ScreenContainer object
var screenCollection = Builder.ScreenContainer;
// adding screen
var screen = screenCollection.AddScreen("NewScreen","full_HD_anthracite");
// saving screenContainer change
screenCollection.Save();
```

#### 19.6.4. AddSymbol, ScreenContainer Method

**Syntax** MoviconNextBuilder.ScreenElement AddSymbol(string screen, string item, string name, double x, double y, double width, double height, [string subfolder = null], [OPCUAViewModel.OPCUAEntityReference tag = null], [bool linkonly = True])

**Description** Adds an element from the symbol library to the indicated screen.

Parameter	Description
string name	element name added
string screen	name of screen to add element to
string item	name of library symbol to be used. The name of the symbols are the names of files contained folder (and subfolders) obtained with the NewScreenTypesFolder function of this class.
double x	x coordinate in pixels
double y	y coordinate in pixels
double width	element's width

double height	element's length
[string subfolder = null]	folder where screen is defined. Optional:
[OPCUAViewModel.OPCUAEntityReference tag = null]	object representing the tag to be connect to the added element. Optional.
[bool linkonly = True]	Insertion flag as link. Optional.

**Result** MoviconNextBuilder.ScreenElement: istanza di un oggetto che rappresenta l'elemento aggiunto allo screen. In caso di errore ritorna null.

**Example:**

```
// instance of ScreenContainer object
var screenCollection = Builder.ScreenContainer;
// adding symbol
var screen = screenCollection.AddSymbol("NewScreen", "Acces_Traffic1"
    , "access"
    , 200, 200, 70, 70);
// saving screenContainer change
screenCollection.Save();
```

#### 19.6.5. AddToolboxItem, ScreenContainer Method

**Syntax** MoviconNextBuilder.ScreenElement AddToolboxItem(string screen, string item, string name, double x, double y, double width, double height, [string subfolder = null], [OPCUAViewModel.OPCUAEntityReference tag = null])

**Description** Adds an element from the toolbox to the indicated screen.

Parameter	Description
string name	Name of added element.
string screen	Name of screen to which to add element.
string item	Name of toolbox element. The names of toolbox elements are the names of files contained in the folder (and subfolders) obtained from the ToolBoxFolder function of this class.
double x	x coordinate in pixels
double y	y coordinate in pixels
double width	element's width
double height	element's height
[string subfolder = null]	folder in which screen is defined. Optional.
[OPCUAViewModel.OPCUAEntityReference tag = null]	Object representing the tag to connect to the added object. Optional.

**Result** MoviconNextBuilder.ScreenElement: instance of an object representing the element added to the screen. Return null when in error.

**Example:**

```
// instance of ScreenContainer object
var screenCollection = Builder.ScreenContainer;
// adding toolbox item
var screen = screenCollection.AddToolboxItem("NewScreen", "Rectangular
CommandButton D" , "Button", 200, 200, 70, 70);
// saving screenContainer change
screenCollection.Save();
```

#### 19.6.6. DeleteElement, ScreenContainer Method

**Syntax** bool DeleteElement(string screen, string elementname, [string subfolder = null])

**Description** Deletes the indicated element from screen.

Parameter	Description
string element name	Name of element to be deleted.
string screen	Name of screen from which element is to be deleted.
[string subfolder = null]	folder in which screen is defined. Optional.

**Result** Boolean

**Example:**

```
// instance of ScreenContainer object
var screenCollection = Builder.ScreensContainer;
// deleting element
screenCollection.DeleteElement("NewScreen","Button");
// saving ScreenContainer change
screenCollection.Save();
```

#### 19.6.7. DeleteScreen, ScreenContainer Method

**Syntax** bool DeleteScreen(string name, [string subfolder = null])

**Description** Deletes the indicated screen

Parameter	Description
string name	Name of the screen to be deleted.
[string subfolder = null]	Folder within which the screen is defined. Optional.

**Result** Boolean

#### Example:

Bool: esito dell'operazione

Esempio

```
// instance of ScreenContainer object
var screenCollection = Builder.ScreensContainer;
// deleting element
screenCollection.DeleteScreen("NewScreen");
// saving ScreenContainer change
screenCollection.Save();
```

### 19.6.8. GetItem, ScreenContainer Method

**Syntax** MoviconNextBuilder.ScreenElement GetItem(string screen, string elementname, [string subfolder = null])

**Description** Gets the instance of an object inserted in the indicated screen.

Parameter	Description
String Scren	screen name
[string subfolder = null]	folder where screen is defined, optional
String elementname	name of the object to be obtained

**Result** MoviconNextBuilder.ScreenElement: Instance of the object present on the screen. Will return null when in error.

#### Example:

Here are some examples of how to obtain objects of different types and how to change some of their properties. In order to use the desired types, however, you will need to insert the necessary references to the assemblies beforehand, for example (not exhaustive):

- ButtonControl: assembly Buttons.dll in the Movicon Next Toolbox folder
- EditDisplay: assembly EditDisplay in the Movicon Next Toolbox folder

- SliderControl: assembly Sliders in the Movicon Next Toolbox folder
- GaugeControl: assembly Gauges in the Movicon Next Toolbox folder

```
// Get TextBox item
var itemToMod = screencollection.GetItem("Screen1", "TextBoxName");
TextBox item = itemToMod.Element as TextBox;
item.Width += 50;
Canvas.SetLeft(item, 500);
Canvas.SetTop(item, 500);
item.Text = "Text in Box";
item.Foreground = Brushes.Blue;
item.Background = Brushes.White;
item.FontSize = 24;
item.FontStyle = FontStyles.Italic;
item.FontWeight = FontWeights.Bold;
item.FontFamily = new FontFamily("Arial");
item.Visibility = Visibility.Visible;
item.Opacity = 0.5;
item.TextAlignment = TextAlignment.Center;
item.TextWrapping = TextWrapping.WrapWithOverflow;
item.ToolTip = "Text Box Tooltip";

// Get ButtonControl item
var itemToMod = screencollection.GetItem("Screen1", "ButtonControlName");
ButtonControl item = itemToMod.Element as ButtonControl;
item.Width += 50;
Canvas.SetLeft(item, 300);
Canvas.SetTop(item, 300);
item.Text = "Button Text";
item.Foreground = Brushes.Red;
item.Background = Brushes.White;
item.FontSize = 24;
item.FontStyle = FontStyles.Italic;
item.FontWeight = FontWeights.Bold;
item.FontFamily = new FontFamily("Arial");
item.Visibility = Visibility.Visible;
item.Opacity = 0.7;
item.TextAlignment = TextAlignment.Center;
item.TextWrapping = TextWrapping.WrapWithOverflow;
item.ToolTip = "Button Control Tooltip";

// Get ToggleButtonControl item
var itemToMod = screencollection.GetItem("Screen1", "ToggleButtonControlName");
ToggleButtonControl item = itemToMod.Element as ToggleButtonControl;
item.Width += 50;
Canvas.SetLeft(item, 300);
Canvas.SetTop(item, 300);
item.Text = "ToggleButton Text";
item.Foreground = Brushes.Red;
item.Background = Brushes.White;
item.FontSize = 24;
item.FontStyle = FontStyles.Italic;
item.FontWeight = FontWeights.Bold;
item.FontFamily = new FontFamily("Arial");
```

```

item.Visibility = Visibility.Visible;
item.Opacity = 0.7;
item.TextAlignment = TextAlignment.Center;
item.TextWrapping = TextWrapping.WrapWithOverflow;
item.ToolTip = "Toggle Button Control Tooltip";

// Get SliderControl item
var itemToMod = screencollection.GetItem("Screen1", "SliderControlName");
SliderControl item = itemToMod.Element as SliderControl;
item.Width += 50;
Canvas.SetLeft(item, 300);
Canvas.SetTop(item, 300);
item.Foreground = Brushes.Red;
item.Background = Brushes.White;
item.FontSize = 10;
item.FontStyle = FontStyles.Italic;
item.FontWeight = FontWeights.Bold;
item.FontFamily = new FontFamily("Arial");
item.Visibility = Visibility.Visible;
item.Opacity = 0.5;
item.MinValue = -200;
item.SliderStyle = Sliders.Enums.SliderStyleEnum.Flat;
item.ShowLabel = true;
item.ToolTip = "Slider Control Tooltip";

// Get GaugeControl item
var itemToMod = screencollection.GetItem("Screen1", "GaugeControlName");
GaugeControl item = itemToMod.Element as GaugeControl;
item.Width += 50;
Canvas.SetLeft(item, 100);
Canvas.SetTop(item, 100);
item.Foreground = Brushes.Red;
item.Background = Brushes.White;
item.FontSize = 10;
item.FontStyle = FontStyles.Italic;
item.FontWeight = FontWeights.Bold;
item.FontFamily = new FontFamily("Arial");
item.Visibility = Visibility.Visible;
item.Opacity = 0.5;
item.MinValue = 50;
item.GaugeBaseModel = Gauges.Enums.PredefinedBaseElementKinds.DarkMetal;
item.MarkerFill = Brushes.AliceBlue;
item.NeedlePresentation = Gauges.Enums.PredefinedElementKinds.Future;
item.ToolTip = "Gauge Control Tooltip";

// Get Display item
var itemToMod = screencollection.GetItem("Screen1", "DisplayName");
EditDisplay.EditDisplay item = itemToMod.Element as EditDisplay.EditDisplay;
item.Width += 50;
Canvas.SetLeft(item, 200);
Canvas.SetTop(item, 50);
item.Foreground = Brushes.Red;
item.Background = Brushes.White;
item.FontSize = 20;

```

```

item.FontStyle = FontStyles.Normal;
item.FontWeight = FontWeights.Bold;
item.FontFamily = new FontFamily("Arial");
item.Visibility = Visibility.Visible;
item.Opacity = 1;
item.TextWrapping = TextWrapping.WrapWithOverflow;
item.ToolTip = "EditDisplay Tooltip";
// needed to save screen with the next call to ProjectBuilder.Save
screen.NeedsSave = true;
// saving the modified configuration
Builder.Save();

```

### Example for symbol elements

Here is an example of how it is possible to manage a symbol item, inserted in the synoptic as incorporated (merged) and not connected (linked). For this purpose, a method for navigating within the symbol must also be inserted to find the desired element, in this example it is called the `GetSymbolElement`. In the example, the assumption is that the symbol, inserted in the synoptic as incorporated (merged) is composed of at least two displays called `EditDisplay_01` and `EditDisplay_02`. Obviously, this can be extended to other types of components, using the appropriate cast, based on the previous examples.

```

// Get Symbol item
var itemToMod = screencollection.GetItem("Screen1", "SymbolName");
EditDisplay.EditDisplay item = GetSymbolElement("EditDisplay_01", "Simbolo03", itemToMod, screen) as
EditDisplay.EditDisplay;
if (item != null) {
item.Width += 50;
item.Foreground = Brushes.Blue;
item.Background = Brushes.White;
item.FontSize = 20;
item.FontStyle = FontStyles.Normal;
item.FontWeight = FontWeights.Bold;
item.FontFamily = new FontFamily("Arial");
item.Visibility = Visibility.Visible;
item.Opacity = 0.5;
item.TextWrapping = TextWrapping.WrapWithOverflow;
item.ToolTip = "EditDisplay Tooltip";
item.MeasUnit = "°C";
item.PrecisionDigits = 3;
}
item = GetSymbolElement("EditDisplay_02", "Simbolo03", itemToMod, screen) as EditDisplay.EditDisplay;
if (item != null) {
item.Width += 50;
item.Foreground = Brushes.Blue;
item.Background = Brushes.Aqua;
item.FontSize = 20;
item.FontStyle = FontStyles.Normal;
item.FontWeight = FontWeights.Bold;
item.FontFamily = new FontFamily("Arial");
item.Visibility = Visibility.Visible;
item.Opacity = 0.5;
item.TextWrapping = TextWrapping.WrapWithOverflow;
item.ToolTip = "EditDisplay Tooltip";
item.MeasUnit = "m";
}

```



```

item.PrecisionDigits = 1;
}
// needed to save screen with the next call to ProjectBuilder.Save
screen.NeedsSave = true;
// saving the modified configuration
Builder.Save();
// needed to help in decoding of merged symbol elements
private UIElement GetSymbolElement(String ElementName, String SymbolName,
MoviconNextBuilder.ScreenElement sE, ScreenDocument sD) {
    foreach (UIElement uie in
((System.Windows.Controls.Canvas)((System.Windows.Controls.Viewbox)sE.Element).Child).Children) {
        if (uie.Uid == ElementName) {
            sD.SubscribePropertyChangeXamlWriterProperties(uie, SymbolName + "-@" + ElementName);
            return uie;
        }
    }
    return null;
}
}

```

#### 19.6.9. GetScreen, ScreenContainer Method

**Syntax**      ScreenSettings.ScreenDocument GetScreen(string name, [string subfolder = null])

**Description**   Gets the object instance representing the indicated screen.

Parameter	Description
string name	screen name
[string subfolder = null]	folder where screen is defined, optional

**Result**      ScreenSettings.ScreenDocument: screen object instance. Returns null if in error.

#### Example:

```

// instance of ScreenContainer object
var screenCollection = Builder.ScreensContainer;
// getting screen
screenCollection.GetScreen("NewScreen");

```

#### 19.6.10. GetScreenUri, ScreenContainer Method

**Syntax**      System.Uri GetScreenUri(string name, [string subfolder = null])

**Description**   Gets the indicated screen's Uri

Parameter	Description
string name	name of screen
[string subfolder = null]	Folder within which the screen is defined. Optional.

**Result** System.Uri: Uri of the desired screen. Returns null if in error.

**Example:**

```
// instance of ScreenContainer object
var screenCollection = Builder.ScreensContainer;
// getting screen Uri
screencollection.GetScreenUri("NewScreen");
```

#### 19.6.11. NewScreenTypesFolder, ScreenContainer Method

**Syntax** string NewScreenTypesFolder()

**Description** Gets the path containing the symbol file contents.

Parameter	Description
-	-

**Result** String

**Example:**

```
// instance of ScreenContainer object
var screenCollection = Builder.ScreensContainer;
// getting screen Uri
var typesFolder = screencollection.NewScreenTypesFolder ();
```

#### 19.6.12. Save, ScreenContainer Method

**Syntax** void Save()

**Description** Saves all the changes pending in the ScreenContainer.

Parameter	Description
-	-

### 19.6.13. Screen Container Examples

You can cast an added screen object with the desired properties in the following way as shown in this example:

## Button

### Content Property:

If the object is the return value from the **AddToolboxItem** function:

```
// instance of ScreenContainer object
var screenCollection = Builder.ScreenContainer;
// add a new toolbox item to the screen
var button = screenCollection.AddToolboxItem("ScreenName", "Rectangular CommandButton D",
"ButtonName", 100, 100, 100, 50);
// to manage some properties of the added object you can use a type casting
((System.Windows.Controls.Button)((System.Windows.Controls.ContentControl)button.Element).Content).Content = "Button Text";
```

Nel caso in cui si debba intervenire sulle proprietà di un oggetto aggiunto in precedenza ad uno screen allora si deve agire in questo modo:

```
// instance of ScreenContainer object
var screenCollection = Builder.ScreenContainer;
// get instance of ScreenDocument object
var screenDoc = screenCollection.GetScreen("ScreenName");
// get instance of ScreenEntity object
var btnEntity = (screenDoc.MapScreenEntities)["ButtonName"];
// get access to the content property by mean of Element object with a cast to the appropriate type
((System.Windows.Controls.Button)((System.Windows.Controls.ContentControl)btnEntity.Element).Content).Content = "Button Text";
```

## TextBox

### FontSize and Text Property:

If the object is the return value from the **AddToolboxItem** function:

```
// instance of ScreenContainer object
var screenCollection = Builder.ScreenContainer;
// add a new toolbox item to the screen
var button = screenCollection.AddToolboxItem("ScreenName", "Text", "TextName", 100, 100, 100, 50);
// to manage some properties of the added object you can use a type casting
((System.Windows.Controls.TextBox)text.Element).FontSize = 15;
((System.Windows.Controls.TextBox)text.Element).Text = "Text Content";
```

#### 19.6.14. ScreenContainer, ScreenContainer Method

**Syntax** ScreenContainer(DocumentManager.ComponentService.IDocument parent, UFProjectManagerComponent manager)

**Description** This is the class builder. A class instance object, that is therefore already build and is created with the initialization procedure described in the initialization phase (see MoviconNextBuilder.ProjectBuilder project instance manager)

Parameter	Description
-	-

**Result** -

#### 19.6.15. SearchXamlItem, ScreenContainer Method

**Syntax** string SearchXamlItem(string itemname, string path)

**Description** Searches for an .xaml file in start folder in a starting folder. This is needed, for example, to get the complete path of an object from the toolbox or symbol library.

Parameter	Description
string itemname	Name of file to search for.
string path	Start path.

**Result** String: complete path of the object searched for.

#### Example:

```
// instance of ScreenContainer object
var screenCollection = Builder.ScreensContainer;
// getting screen Uri
var itemPath = screenCollection.SearchXamlItem("AlarmWindowControl",
"C:\\ProgramData\\Progea\\Movicon.NExT.3.1")
```

#### 19.6.16. SetBackground, ScreenContainer Method

**Syntax** bool SetBackground(string name, System.Windows.Media.Brush background, [string subfolder = null])

**Description** Sets the indicated screen background.

Parameter	Description
string name	Name of screen.
System.Windows.Media.Brush background	System.Window.Media.Brushes enumeration element.
[string subfolder = null]	Folder in which screen is defined. Optional.

**Result** Boolean

**Example:**

```
// instance of ScreenContainer object
var screenCollection = Builder.ScreensContainer;
// deleting element
screenCollection.SetBackground("NewScreen" , Brushes.Black);
// saving ScreenContainer change
screenCollection.Save();
```

#### 19.6.17. SetScreenEntity, ScreenContainer Method

**Syntax** bool SetScreenEntity(string screen, string name, OPCUAViewModel.OPCUAEntityReference tag, [string subfolder = null])

**Description** sets the Tag Item for the nameobject name in the screenname screen.

Parameter	Description
<b>String screen</b>	name of screen
<b>String name</b>	name of object in screen
<b>OPCUAViewModel.OPCUAEntityReference tag</b>	refers to the tag to be used in the object
<b>[string subfolder = null]</b>	folder within which the optional screen is defined.

**Result** Boolean

**Example:**

```
// instance of ScreenContainer object
var screenCollection = Builder.ScreensContainer;
```

```

var server = Builder.IODataServer
var tagScreen = server.GetTagEntityReference("NomeTag");

// setting screen element tag
screenCollection.SetScreenEntity(screenName, "editVar", tagScreen, "screenFolder");
// saving ScreenContainer change
screenCollection.Save();

```

### 19.6.18. ToolBoxFolder, ScreenContainer Method

**Syntax**      string ToolBoxFolder()

**Description**   Gets the path in which the toolbox element files are contained.

Parameter	Description
-	-

**Result**      String: Toolbox file path

#### Example:

```

// instance of ScreenContainer object
var screenCollection = Builder.ScreensContainer;
// getting screen Uri
var typesFolder = screenCollection.ToolBoxFolder();

```

## 19.7. IODataServer

### 19.7.1. AddAlarmArea, IODataServer Method

**Syntax**      UFUAModel.UFUAArea AddAlarmArea(string name, [UFUAModel.UFUAArea parent = null])

**Description**   Adds an Alarm Area with name in the parent Alarm area.

Parameter	Description
string name	name of new alarm area
UFUAModel.UFUAArea parent	reference to eventual parent alarm area (optional).

**Result**      UFUAModel.UFUAArea (object representing the alarm area just added)

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// add a new Alarm Definition
// add alarm area
var alarmArea = server.AddAlarmArea("NewAlarmArea");
// add a new Alarm Definition in the area
var subarea = server.AddAlarmArea("NewAlarmSubArea", alarmArea);
// saving IOServer
```

### 19.7.2. AddAlarmDefinition, IODataServer Method

**Syntax**      UFUAModel.UFUAAAlarmDefinition AddAlarmDefinition(string name,  
UFUAModel.UFUAAAlarmSource parent, [bool overwrite = False])

**Description**    Adds a new definition to the alarm with name of 'name' within the source. .

Parameter	Description
string name	name of the new alarm definition
UFUAModel.UFUAAAlarmSource parent	reference to the alarm source in which to define the new alarm
bool overwrite	overwrites an existing alarm

**Result**      UFUAModel.UFUAAAlarmDefinition (object representing the alarm definition just added)

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// add a new Alarm Definition
// add alarm area
var alarmArea = server.AddAlarmArea("NewAlarmArea");
// add alarm source in the area
var source = server.AddAlarmSource("NewAlarmSource", alarmArea);
// add alarm edfinition for the source
var alarm = server.AddAlarmDefinition("NewAlarm", source);
// saving IOServer configuration change
server.Save();
```

### 19.7.3. AddAlarmSource, IODataServer Method

**Syntax**      UFUAModel.UFUAAAlarmSource AddAlarmSource(string name, UFUAModel.UFUAArea parent)

**Description**    Adds a new alarm name definition within the source.

Parameter	Description
string name	name of the new alarm source
UFUAModel.UFUAArea parent	reference to the area in which to define new source

**Result**      UFUAModel.UFUAAAlarmSource (object represening the alarm source just added)

#### Example:

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// add a new Alarm Definition
// add alarm area
var alarmArea = server.AddAlarmArea("NewAlarmArea");
// add alarm source in the area
var source = server.AddAlarmSource("NewAlarmSource", alarmArea);
// saving IOSever configuration change
server.Save();
```

### 19.7.4. AddBaseAddress, IODataServer Method

**Syntax**      UFUAModel.UFUABaseAddress AddBaseAddress([string transport = net.pipe])

**Description**    Adds tranport type.

Parameter	Description
string transport	name of new transport, possible values: <ul style="list-style-type: none"><li>• net.pipe</li><li>• net.tcp</li><li>• opc.tcp</li><li>• http</li><li>• https</li><li>• nosecurityhttp</li></ul>

**Result**      UFUAModel.UFUABaseAddress: object representing the transport just added.

#### Example:



```
// instance of IODataServer variable
var server = Builder.IODataServer;
// add the transport net.tcp to the IOServer
var baseAddress = server.AddBaseAddress("net.tcp");
// changing a property of the transport object
baseAddress.Port = 62847;
// saving IOServer configuration change
server.Save();
```

### 19.7.5. AddDatalogger, IODataServer Method

**Syntax** DataLoggerModel.DataLoggerSettings AddDatalogger(string name, [bool overwrite = False])

**Description** Adds a DataLogger Structure with name as name

Parameter	Description
string name	name of DataLogger Structure to be added.
[bool overwrite = False]	Overwrites an existing Structure, optional. In the presense of a datalogger homonym, the return value will be null if no overwrite is requested.

**Result** DataLoggerModel.DataLoggerSettings: object that consents access to the various properties of the Datalogger Structure just inserted.

#### Example:

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting aa existing tag reference
var tag = server.GetTag("TagName");
if (tag != null) {
    // adding a DataLogger
    var dl = server.AddDatalogger("DataLoggerName");

    if (dl != null) {
        // enabling the DataLogger
        dl.Enable = true;
        // setting the sampling time (in this example the interval time is represented in hundred nanoseconds
        // (10000 ticks in a millisecond)
        dl.RecordingTimeInterval = new TimeSpan(100000000); // 1 sec of interval
        // adding a DataLogger New column
        var col = server.AddDataLoggerColumn("DataLoggerColumn", dl);
        // adding a tag reference to the new added column
        if (tag != null)
            col.ColumnTag = new UFUAModel.TagEntityReference(tag);
    }
}
```

```
// saving IO Server configuration change
server.Save();
```

#### 19.7.6. AddDataLoggerColumn, IODataServer Method

**Syntax**      `DataLoggerModel.DataLoggerColumn AddDataLoggerColumn(string name, DataLoggerModel.DataLoggerSettings datalogger, [bool overwrite = False])`

**Description**    Consents to adding a new column definition in the datalogger

Parameter	Description
string name	name of the Column to be added
DataLoggerModel.DataLoggerSettings datalogger	datalogger to which the new column definition is to be added
[bool overwrite = False]	overwrites an existing column, optional. In cases of homonomous columns, the return value will be null unless an overwrite is requested.

**Result**      `DataLoggerModel.DataLoggerColumn`: object which consents acced to the various added column's properties.

#### Example:

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting aa existing tag reference
var tag = server.GetTag("TagName");
if (tag != null) {
    // adding a DataLogger
    var dl = server.AddDatalogger("DataLoggerName");

    if (dl != null) {
        // enabling the DataLogger
        dl.Enable = true;
        // setting the sampling time (in this example the interval time is represented in hundred nanoseconds
        // 10000 ticks in a millisecond)
        dl.RecordingTimeInterval = new TimeSpan(100000000); // 1 sec of interval
        // adding a DataLogger New column
        var col = server.AddDataLoggerColumn("DataLoggerColumn", dl);
        // adding a tag reference to the new added column
        if (tag != null)
            col.ColumnTag = new UFUAModel.TagEntityReference(tag);
    }
}
// saving IO Server configuration change
server.Save();
```

### 19.7.7. AddEngineeringUnit, IODataServer Method

**Syntax**      UFUAModel.UFUAEngineeringUnit AddEngineeringUnit(string name, [bool overwrite = False])

**Description**    Adds the engineering unit name

Parameter	Description
string name	name of the engineering unit to be added
[bool overwrite = False]	overwrites existing unit, optional. In cases of an homonymous unit the return value will be null if overwrite is not requested.

**Result**      UFUAModel.UFUAEngineeringUnit: object that consents access to the various properties of the added engineering unit.

#### Example:

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// engineering unit creation
var cust = server.AddEngineeringUnit("customUnit");
if (cust != null) {
    cust.EURangeHigh = 1000;
    cust.EURangeLow = 0;
    cust.InstrumentRangeHigh = 10000;
    cust.InstrumentRangeLow = 10;
    cust.UnitName = "twix";
}
// saving IODataServer configuration change
server.Save();
```

### 19.7.8. AddFolder, IODataServer Method

**Syntax**      UFUAModel.UFUAFolder AddFolder(string name, [UFUAModel.UFUAFolder folder = null])

**Description**    Used to add a folder to a specific tree level of the variables in the IO DataServer.

Parameter	Description
string name	name of the Folder to be added
[bool overwrite = False]	folder in which to insert the new folder. Optional, if omitted the folder will be added at the root of the tag tree.

**Result**      UFUAModel.UFUAFolder: object that consents access to the various added folder's properties.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// add a folder in the tag tree root
var tagFolder = server.AddFolder("Folder");
// add a subfolder in the tag tree
tagFolder = server.AddFolder("Folder2", tagFolder);
// saving IODataServer configuration change
server.Save();
```

**19.7.9. AddHistorian, IODataServer Method**

**Syntax**      UFUAModel.UFUAHistorianSettings AddHistorian(string name, [bool overwrite = False])

**Description**    Used to add a new Historian definition.

Parameter	Description
string name	name of Historian to be added
[bool overwrite = False]	overwrites an existing Historian, optional. In cases of homonymous Structure, the return value will be null if no overwrite is requested.

**Result**      UFUAModel.UFUAHistorianSettings: object which allows access to the various properties of the added Historian .



In the example which follows, you will also see how to add a tag to the Historian just created.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting an existing tag reference
var tag = server.GetTag("TagName");
if (tag != null) {
    var historianName = "NewHistorian";
    // new Historian
    var historian = server.AddHistorian(historianName);
    if (historian != null) {
        // some properties of Historian
        historian.ExceptionDeviationFormat = UFUAModel.DeviationType.AbsoluteValue;
        historian.ExceptionDeviation = 1.0;
        historian.MaxTimeInterval = new TimeSpan(0, 0, 2); // 1 sec of inreval
        historian.MinTimeInterval = new TimeSpan(0, 0, 2); // 1 sec of inreval
    }
}
```

```

}
// assign historian reference to the tag
tag.HistorianSettings = historianName;
// saving IO Server configuration change
server.Save();

```

#### 19.7.10. AddNewDriver, IODataServer Method

**Syntax**      `bool AddNewDriver(string assemblyname, string friendlyname, string factory)`

**Description**    Used to add a driver to the IO DataServer.

Parameter	Description
String assemblyname	name of driver dll
String friendlyname	mnemonic reference name to driver in the Movicon NeXT project
String factory	name of builder

**Result**      Boolean

#### Example:

```

// instance of IODataServer variable
var server = Builder.IODataServer;
// add the ModbusTcp IO Driver
server.AddNewDriver("ModbusTCP.dll", "ModbusTcp", "Modbus");
// saving IO Server configuration change

```

#### 19.7.11. AddPrototype, IODataServer Method

**Syntax**      `UFUAModel.UFUATagPrototype AddPrototype(string name, [bool overwrite = False])`

**Description**    Adds a data model Structure to the IO DataServer.

Parameter	Description
String name	name of the data model Structure to be added
[bool overwrite = False]	overwrites an existing Structure, optional. In the presence of an homonymous Structure, the value will return null if not requested to overwrite.

**Result**      UFUAModel.UFUATagPrototype : object that allows access to the various properties of the added data model Structure.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// prototype creation
string prototypeName = "structVar";
var proto = server.AddPrototype(prototypeName);
// saving IOSever configuration change
server.Save();
```

### 19.7.12. AddPrototypeMember, IODataServer Method

**Syntax**      UFUAModel.UFUATag AddPrototypeMember(string name,  
UFUAModel.UFUATagPrototype proto, [UFUAModel.DataType type = 2], [bool  
overwrite = False])

**Description**    Adds a member to an existing data model Structure.

Parameter	Description
String name	name of member to be added.
UFUAModel.UFUATagPrototype proto	data model Structure to add member to.
[UFUAModel.DataType type = 2]	member's data type. The data types allowed are those deinge in the UFUAModel.DataType enumeration: <ul style="list-style-type: none"><li>• Boolean</li><li>• Byte</li><li>• Double</li><li>• Float</li><li>• Int16</li><li>• Int32</li><li>• Int64</li><li>• Sbyte</li><li>• String</li><li>• UInt16</li><li>• UInt32</li><li>• UInt64</li></ul>
[bool overwrite = False]	overwrites an existing member, optional. In cases where a Structure of the same name exists, the value will return null if not requested to overwrite.

**Result**      UFUAModel.UFUATag : object allowing access to the various properties of the added member.



In the following example you will also see how to a create a tag from the data model Structure just created.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// prototype creation
string prototypeName = "structVar";
var proto = server.AddPrototype(prototypeName);
// add prototype members
if (proto != null) {
    server.AddPrototypeMember("Indication", proto, UFUAModel.DataType.Int16);
    server.AddPrototypeMember("Label", proto, UFUAModel.DataType.String);
    server.AddPrototypeMember("Distance", proto, UFUAModel.DataType.Double);
    server.AddPrototypeMember("Present", proto, UFUAModel.DataType.Boolean);
    server.AddPrototypeMember("Quantity", proto, UFUAModel.DataType.Int32);
}
// prototype element tag creation
var tag = server.AddTag("Element");
if (tag != null) {
    tag.ModelType = UFUAModel.ModelType.ObjectType;
    tag.PrototypeModel = prototypeName;
    tag.Description = "Production element";
}
// saving IOServer configuration change
server.Save();
```

**19.7.13. AddStation, IODataServer Method**

**Syntax**      object AddStation(string assemblyname, string channelname)

**Description**    Adds a station to the indicated channel.

Parameter	Description
String assemblyname:	name of the driver's dll
String channelname	name of the channel

**Result**      object: reference to the instance of the added station, a cast must be performed to the desired station type, based on the driver. It will return null in case of bankruptcy.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// add the ModbusTcp IODriver
var nStat = server.AddStation("ModbusTCP.dll","Chan01") as ModbusTCPStationSettings;
// saving IOServer configuration change
```

```
server.SaveDriver("ModbusTCP.dll");
```

#### 19.7.14. AddTag, IODataServer Method

**Syntax**      `UFUAModel.UFUATag AddTag(string name, [UFUAModel.UFUAFolder folder = null], [UFUAModel.DataType type = 2], [bool overwrite = False])`

**Description**    Used to define a new tag and to add it to the IO DataServer

Parameter	Description
String name	Name of tag to be added.
UFUAModel.UFUAFolder folder	Folder in which to add tag, optional.
[UFUAModel.DataType type = 2]	Tag's data type, optional. The data types allows are those defined in the UFUAModel.DataType enumeration: <ul style="list-style-type: none"><li>• Boolean</li><li>• Byte</li><li>• Double</li><li>• Float</li><li>• Int16</li><li>• Int32</li><li>• Int64</li><li>• Sbyte</li><li>• String</li><li>• UInt16</li><li>• UInt32</li><li>• UInt64</li></ul>
[bool overwrite = False]	Overwrites an existing tag, optional. When a tag with the same name exists already, the value will return null unless requested to overwrite it.

**Result**      `UFUAModel.UFUATag`: object allowing access to the various added tag's properties.

#### Example:

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// integer tag creation
tag = server.AddTag("NewIntegerTag", null, UFUAModel.DataType.UInt16, true);
if (tag != null) {
    tag.ArrayDimension = 0;
    tag.Description = "New integer tag";
    tag.InitialValue = "0";
}
// add a folder in the tag tree root
var tagFolder = server.AddFolder("Folder");
// boolean tag creation into the new folder
tag = server.AddTag("NewBooleanTag", tagFolder, UFUAModel.DataType.Boolean, true);
```



```

if (tag != null) {
    tag.ArrayDimension = 0;
    tag.Description = "New boolean tag";
    tag.InitialValue = "false";
}
// saving IOserver configuration change
server.Save();

```

#### 19.7.15. AssignAlarmToTag, IODataServer Method

**Syntax**      `bool AssignAlarmToTag(UFUAModel.UFUATag tag, UFUAModel.UFUAAAlarmDefinition alarm, [int type = 0], [string text = ])`

**Description**    Assigns an existing Alarm Definition to the tag.

Parameter	Description
UFUAModel.UFUATag tag	Tag to which the Alarm Definition is to be assigned.
UFUAModel.UFUAAAlarmDefinition alarm	Alarm Definition to be assigned.
[int type = 0]	Assignment mode, optional <ul style="list-style-type: none"> <li>• 0: Single</li> <li>• 1: AnyTagBit</li> <li>• 2: AntTagElement (for tag array)</li> </ul>
[string text = ]	alarm text, optional

**Result**      -

#### Example:

```

// instance of IODataServer variable
var server = Builder.IODataServer;
// getting a tag reference from IOserver
var tag = server.GetTag("TagName");
// getting an alarm reference from IOserver
var alarm = server.GetAlarmDefinition("NewAlarm", server.GetAlarmSource("NewAlarmSource",
server.GetAlarmArea("NewAlarmArea")));
if (tag != null && alarm != null)
    // assign an alarm to the tag
    server.AssignAlarmToTag(tag, alarm, 0, "New Alarm text");
// saving IOserver configuration change
server.Save();

```

#### 19.7.16. DeleteAlarmArea, IODataServer Method

**Syntax**      `bool DeleteAlarmArea(string name, [UFUAModel.UFUAAArea parent = null])`

**Description** Used to delete alarm area definition

Parameter	Description
string name	Name of alarm area to delete.
[UFUAModel.UFUAArea parent = null]	Any area in which the area to be deleted is defined, optional.

**Result** Boolean



This function will delete the alarm area even when empty.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// delete alarm area
server.DeleteAlarmArea("NewAlarmArea");
// saving IOServer configuration change
server.Save();
```

### 19.7.17. DeleteAlarmDefinition, IODataServer Method

**Syntax** bool DeleteAlarmDefinition(string name, UFUAModel.UFUAAAlarmSource parent)

**Description** Used to delete alarm definition

Parameter	Description
string name	name of alarm definition to be deleted
UFUAModel.UFUAAAlarmSource parent	name of the alarm source containing the alarm definition

**Result** Boolean



This function will delete the alarm definition even when already assigned a tag.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// get the alarm area instance
var area = server.GetAlarmArea("NewAlarmArea");
// get the alarm source instance
```

```

var source = server.GetAlarmSource("NewAlarmSource", area);
// delete the alarm definition
server.DeleteAlarmDefinition("NewAlarm", source);
// saving IO Server configuration change
server.Save();

```

### 19.7.18. DeleteAlarmSource, IODataServer Method

**Syntax**      `bool DeleteAlarmSource(string name, UFUAModel.UFUAArea parent)`

**Description**    Deletes the definition of the alarm source

Parameter	Description
string name	name of the alarm source to be deleted
UFUAModel.UFUAArea parent	alarm area containing the source to be deleted

**Result**      Boolean



This function will delete the alarm source even when empty.

#### Example:

```

// instance of IODataServer variable
var server = Builder.IODataServer;
// get the alarm area instance
var area = server.GetAlarmArea("NewAlarmArea");
// delete alarm source
server.DeleteAlarmSource("NewAlarmSource", area);
// saving IO Server configuration change
server.Save();

```

### 19.7.19. DeleteDatalogger, IODataServer Method

**Syntax**      `bool DeleteDatalogger(string name)`

**Description**    Deletes a DataLogger

Parameter	Description
string name	name of DataLogger to be deleted

**Result** Boolean



This function will delete data logger even when empty.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// delete DataLogger
server.DeleteDatalogger("DataLoggerName");
// saving IOserver configuration change
server.Save();
```

### 19.7.20. DeleteDataloggerColumn, IODataServer Method

**Syntax** bool DeleteDataloggerColumn(string name,  
DataLoggerModel.DataLoggerSettings datalogger)

**Description** Deletes the definition of a column in the indicated DataLogger

Parameter	Description
string name	name of column to be deleted
DataLoggerModel.DataLoggerSettings datalogger	DataLogger from which column is to be deleted

**Result** Boolean

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting DataLogger instance
var dataLogger = server.GetDataLogger("DataLoggerName");
// delete DataLogger column
server.DeleteDataloggerColumn("DataLoggerColumnName", dataLogger);
// saving IOserver configuration change
server.Save();
```

### 19.7.21. DeleteEngineeringUnit, IODataServer Method

**Syntax** bool DeleteEngineeringUnit(string name)

**Description** Deletes indicated engineering unit

Parameter	Description
string name	name of engineering unit to be deleted

**Result** Boolean

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// delete Engineering Unit
server.DeleteEngineeringUnit("customUnit");
// saving IODataServer configuration change
server.Save();
```

### 19.7.22. DeleteFolder, IODataServer Method

**Syntax** bool DeleteFolder(string name, [UFUAModel.UFUAFolder folder = null])

**Description** Deletes the folder indicated by the folder passed as second parameter. If the second parameter is not indicated, the folder will be cancelled from the root.

Parameter	Description
string name	Name of folder to be deleted.
[UFUAModel.UFUAFolder folder = null]	Name of any eventual folder containing the folder to be deleted, if allowed the folder will be deleted from the root. To obtain the folder object instance, use IODataServer.GetFolder, optional.

**Result** Boolean



This function will delete the folder even when empty.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// delete Folder
server.DeleteFolder("Folder2",server.GetFolder("Folder"));
// saving IODataServer configuration change
server.Save();
```

### 19.7.23. DeleteHistorian, IODataServer Method

**Syntax**      `bool DeleteHistorian(string name, [bool removefromtag = False])`

**Description**    Deletes the indicated Historian removing any tag associations.

Parameter	Description
string name	Name of Historian to be deleted.
[bool removefromtag = False]	When set to True, removes any associations to the Historian of previously assigned tags. Optional.

**Result**      Boolean

#### Example:

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// delete Historian
server.DeleteHistorian("NewHistorian");
// saving IODataServer configuration change
server.Save();
```

### 19.7.24. DeletePrototype, IODataServer Method

**Syntax**      `bool DeletePrototype(string name)`

**Description**    Deletes the indicated tag Structure.

Parameter	Description
string name	Name of Structure to be deleted.

**Result**      Boolean



This function deletes the Structure even when empty and/or if there are tags based on it.

#### Example:

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// delete Prototype
server.DeletePrototype("structVar");
// saving IODataServer configuration change
server.Save();
```

### 19.7.25. DeleteTag, IODataServer Method

**Syntax**      `bool DeleteTag(string name, [UFUAModel.UFUAFolder folder = null])`

**Description**      Deletes indicated Tag.

Parameter	Description
string name	Name of tag to be deleted.
[UFUAModel.UFUAFolder folder = null]	Instance of the folder object containing the tag. If allowed, it will be deleted from the root. To get the folder object's instance you must use <code>IODataServer.GetFolder</code> . Optional.

**Result**      Boolean

#### Example:

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting the tag folder
var tagFolder server.GetFolder("tagFolder");
// delete Tag
server.DeleteTag("tagName", tagFolder);
// saving IODataServer configuration change
server.Save();
```

### 19.7.26. GetAlarmArea, IODataServer Method

**Syntax**      `UFUAModel.UFUAArea GetAlarmArea(string name, [UFUAModel.UFUAArea parent = null])`

**Description**      Gets the instance of a `UFUAModel.UFUAArea` type object representing the indicated alarm area.

Parameter	Description
string name	Name of Alarm Area to retrieve.
[UFUAModel.UFUAFolder folder = null]	Instance of the alarm area containing the area to be retrieved. If the alarm area has been added, it will be contained at the root. Optional.

**Result**      `UFUAModel.UFUAArea`: instance of the desired AlarmArea object. When in error, will return null.

#### Example:

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting the alarm area
var area server.GetAlarmArea("NewAlarmArea");
```

### 19.7.27. GetAlarmDefinition, IODataServer Method

**Syntax**      UFUAModel.UFUAAAlarmDefinition GetAlarmDefinition(string name, UFUAModel.UFUAAAlarmSource parent)

**Description**    Gets the instance of an UFUAModel.UFUAAAlarmDefinition object representing the indicated alarm definition.

Parameter	Description
string name	Name of the alarm definition to be retrieved.
UFUAModel.UFUAAAlarmSource parent	Instance of the Alarm Source object in which the alarm is defined.

**Result**      UFUAModel.UFUAAAlarmDefinition: Instance of the desired alarm definition. When in error returns null.

#### Example:

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting the alarm
var alarm server.GetAlarmDefinition("NewAlarm", server.GetAlarmSource("NewAlarmSource", server.GetAlarmArea("NewAlarmArea")));
```

### 19.7.28. GetAlarmSource, IODataServer Method

**Syntax**      UFUAModel.UFUAAAlarmSource GetAlarmSource(string name, UFUAModel.UFUAAArea parent)

**Description**    Gets the instance of an UFUAModel.UFUAAAlarmSource object representing the indicated alarm source.

Parameter	Description
string name	Name of the desired alarm source.
UFUAModel.UFUAAArea parent	Instance of the alarm area in which source is defined.

**Result**      UFUAModel.UFUAAAlarmSource: Instance of the alarm source desired. Returns null when in error.



**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting the alarm source
var source server.GetAlarmSource("NewAlarmSource", server.GetAlarmArea("NewAlarmArea"));
```

**19.7.29. GetBaseAddress, IODataServer Method**

**Syntax**      UFUAModel.UFUABaseAddress GetBaseAddress(string transport)

**Description**   Gets the instance of the UFUAModel.UFUABaseAddress object representing the indicated transport.

Parameter	Description
string transport	transport type to retrieve.

**Result**      UFUAModel.UFUABaseAddress: indicated transport object instance. When in error, returns null.



Returned object can be used to delete transport type.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting the base address
var transport server.GetBaseAddress("net.tcp");
// deleting transport object
transport.Delete();
// saving IO Server configuration change
server.Save();
```

**19.7.30. GetChannel, IODataServer Method**

**Syntax**      object GetChannel(string assemblyname, string channelname)

**Description**   Gets a reference to the indicated channel for the driver.

Parameter	Description
String assemblyname:	name of the driver's dll
String channelname	name of the channel

**Result**      object: reference to the instance of the indicated channel, a cast must be performed to the desired channel type, based on the driver. If the channel does not exist it will return null.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// remove a ModbusTcp driver channel
var nChan = server.GetChannel("ModbusTCP.dll","Chan01") as ModbusTCPChannelSettings;
nChan.WaitTime = 200;
// saving IOServer configuration change
server.SaveDriver("ModbusTCP.dll");
```

### 19.7.31. GetDataLogger, IODataServer Method

**Syntax**      DataLoggerModel.DataLoggerSettings GetDataLogger(string name)

**Description**   Gets the indicated DataLoggerModel.DataLoggerSettings object instance.

Parameter	Description
string name	Name of DataLogger to retrieve.

**Result**      DataLoggerModel.DataLoggerSettings: indicated DataLogger object instance. When in error returns Null.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting the data logger
var dataLogger server.GetDataLogger("DataLoggerName");
```

### 19.7.32. GetDataLoggerColumn, IODataServer Method

**Syntax**      DataLoggerModel.DataLoggerColumn GetDataLoggerColumn(string name, DataLoggerModel.DataLoggerSettings datalogger)

**Description**   Gets instance of the indicated DataLoggerModel.DataLoggerColumn column object belonging to the indicated DataLoggerModel.DataLoggerSettings Datalogger.

Parameter	Description
string name	Name of the column to be retrieved
DataLoggerModel.DataLoggerSettings datalogger	Instance of the Datalogger object which column belongs to.

**Result** DataLoggerModel.DataLoggerColumn: indicated column object instance. When in error returns null.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting the data logger
var dataLogger server.GetDatalogger("DataLoggerName");
// getting the data logger column
var dataLoggerColumn = server.GetDataloggerColumn("DataLoggerName", dataLogger);
```

### 19.7.33. GetDataloggerList, IODataServer Method

**Syntax** System.Collections.Generic.List<string> GetDataloggerList()

**Description** Gets the list of all the Dataloggers configured in the project.

Parameter	Description
-	-

**Result** System.Collections.Generic.List<string>: DataLogger list.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;

// getting the data logger
var DataloggerList server.GetDataloggerList();
```

### 19.7.34. GetDriverConfiguration, IODataServer Method.htm

**Syntax** object GetDriverConfiguration(string assemblyname)

**Description** Gets the instance of the driver configuration object indicated as a parameter. The returned object is generic, and will then be specified by means of a cast. If the driver does not exist it will return null.

Parameter	Description
String assemblyname:	name of the driver's dll

**Result**      Object: instance of the driver configuration object.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting the driver configuration
var drvConf = Builder.IODataServer.GetDriverConfiguration("ModbusTCP.dll");
// casting the returned object in order to modify some property
((ModbusTCPDriverSettings)drvConf).AggregationThreshold = 5;
server.SaveDriver("ModbusTCP.dll");
```

### 19.7.35. GetEngineeringUnit, IODataServer Metod

**Syntax**      UFUAModel.UFUAEngineeringUnit GetEngineeringUnit(string name)

**Description**    Gets instance of the desired UFUAModel.UFUAEngineeringUnit object.

Parameter	Description
string name	name of engineering unit to be retrieved.

**Result**      DataLoggerModel.DataLoggerColumn: indicated column object instance. When in error, returns null.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting the engineering unit
var engUnit server.GetEngineeringUnit("customUnit");
```

### 19.7.36. GetFolder, IODataServer Metod

**Syntax**      UFUAModel.UFUAFolder GetFolder(string name, [UFUAModel.UFUAFolder folder = null])

**Description**    Gets an instance of the desired UFUAModel.UFUAFolder object if contained in another folder.

Parameter	Description
string name	Name of folder to retrieve.
[UFUAModel.UFUAFolder folder = null]	Any eventual object folder instance containing the desired folder.

**Result**      UFUAModel.UFUAFolder: instance object of indicated folder. Returns null when in error.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// get folder
var folder = server.GetFolder("Folder2",server.GetFolder("Folder"));
// saving IOserver configuration change
server.Save();
```

### 19.7.37. GetHistorian, IODataServer Method

**Syntax**      UFUAModel.UFUAHistorianSettings GetHistorian(string name)

**Description**    Gets an instance from the indicated UFUAModel.UFUAHistorianSettings object.

Parameter	Description
string name	name of historian to be retrieved

**Result**      UFUAModel. UFUAHistorianSettings: indicated Historian's instance object. Returns null when in error.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting Historian
var historian = server.GetHistorian("NewHistorian");
```

### 19.7.38. GetHistorianList, IODataServer Method

**Syntax**      System.Collections.Generic.List<string> GetHistorianList()

**Description**    Gets the list of all the Historians configured in the project.

Parameter	Description
-	-

**Result**      System.Collections.Generic.List<string>: Historian list.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;

// getting the historian list
var HistorianList server.GetHistorianList();
```

**19.7.39. GetPrototype, IODataServer Method**

**Syntax**      UFUAModel.UFUATagPrototype GetPrototype(string name)

**Description**   Gets the instance of the indicated UFUAModel.UFUATagPrototype object.

Parameter	Description
string name	name of prototype to retrieve.

**Result**      UFUAModel.UFUATagPrototype : instance object of indicated prototype. When in error, returns null.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting Prototype
var proto = server.GetPrototype("structVar");
```

**19.7.40. GetServerConfiguration, IODataServer Method**

**Syntax**      UFUAModel.UFUAConfiguration GetServerConfiguration()

**Description**   gets an instance from the UFUAModel.UFUAConfiguration object of the IODataServer server.

Parameter	Description
-	-

**Result**      UFUAModel.UFUAConfiguration: instance object of the IODataServer server. When in error returns null.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting server configuration
```

```
var serverConfig = server.GetServerConfiguration();
```

#### 19.7.41. GetServerEntityReference, IODataServer Method

**Syntax** OPCUAViewModel.OPCUAEntityReference GetServerEntityReference()

**Description** Gets an instance from the OPCUAViewModel.OPCUAEntityReference object of the IODataServer server.

Parameter	Description
-	-

**Result** OPCUAViewModel.OPCUAEntityReference: instance object of the IODataServer server. In case of error returns null.

#### Example:

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting server entity reference
var serverReference = server.GetServerEntityReference();
```

#### 19.7.42. GetStation, IODataServer Method.htm

**Syntax** object GetStation(string assemblyname, string channelname, string stationname)

**Description** Gets the station indicated for the channel in the driver.

Parameter	Description
String assemblyname:	name of the driver's dll
String channelname	Channel Name
String stationname	Station Name

**Result** object: reference to the present station instance, a cast must be performed to the desired station type, based on the driver. If the station is not present it will return null.

#### Example:

```
// instance of IODataServer variable
var server = Builder.IODataServer;
```

```
// add the ModbusTcp IODriver
var nStat = server.GetStation("ModbusTCP.dll","Chan01","Stat0") as ModbusTCPStationSettings;
// saving IO Server configuration chang
```

#### 19.7.43. GetTag, IODataServer Method

**Syntax**      UFUAModel.UFUATag GetTag(string name, [UFUAModel.UFUAFolder folder = null])

**Description**   Gets the UFUAModel.UFUATag object's instance of the indicated tag in the folder it belongs to.

Parameter	Description
string name	name of tag to be retrieved
[UFUAModel.UFUAFolder folder = null]	any eventual folder object instance containing the desired tag.

**Result**      UFUAModel.UFUATag: the desired tag's instance object. Returns null when in error.

#### Example:

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting the tag folder
var tagFolder server.GetFolder("tagFolder");
// getting Tag
var tag = server.GetTag("tagName", tagFolder);
```

#### 19.7.44. GetTagEntityReference, IODataServer Method

**Syntax**      OPCUAViewModel.OPCUAEntityReference GetTagEntityReference(string name, [string instance = null])

**Description**   Gets an instance of the indicated tag's OPCUAViewModel.OPCUAEntityReference object. Attention, this method only functions when the Save method has been called after having added the tag.

Parameter	Description
string name	name of tag to retrieve, possibly composed according to the folder in which it is contained (eg. Folder\varname)
[string instance = null]	instance name when tag is a prototype type.

**Result**      OPCUAViewModel.OPCUAEntityReference: desired tag's instance object. Return null when in error.





The object returned by the server function as parameter in objects belonging to the toolbox which have been inserted in a screen.

#### Example:

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting Tag
var tagReference = server.GetTagEntityReference("tagName");
// getting Proto Tag
var tagReference = server.GetTagEntityReference("memberName","tagName");
```

**Syntax** OPCUAViewModel.OPCUAEntityReference  
**(overloaded)** GetTagEntityReference(UFUAModel.UFUATag tag)  
**Description** Ottiene un'istanza dell'oggetto OPCUAViewModel.OPCUAEntityReference della tag indicata.

Parameter	Description
UFUAModel.UFUATag Tag:	UFUAModel.UFUATag object type that represents the tag. An object of this type can be retrieved by calling the GetTag, or as a result of an Addtag. This method also functions for tag recently added to the server.

**Result** OPCUAViewModel.OPCUAEntityReference: instance object of the tag desired. When in error returns null.

#### Example:

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting the tag folder
var tagFolder server.GetFolder("tagFolder");
// getting Tag
var tag = server.GetTag("tagName", tagFolder);
// getting Tag
var tagReference = server.GetTagEntityReference(tag);
// getting Proto Tag
tag = server.GetUFATag("tagName", "memberName");
// getting Proto Tag
tagReference = server.GetTagEntityReference(tag);
```

#### 19.7.45. GetTagOPCUAEntityReference, IODataServer Method

**Syntax** OPCUAViewModel.OPCUAEntityReference GetTagOPCUAEntityReference(string tagname, [string membername = null])

**Description** Gets an instance from the OPCUAViewModel.OPCUAEntityReference object of the tag indicated, if structure type.

Parameter	Description
string tagname	name of tag to be retrieved, eventually composed according to the folder in which it is contained. (eg. Folder\varname)
[string memberinstance = null]	eventual structure member name, if the tag is structure type. The syntax must mirror the prototype structure (eg. membername or folder\membername or substructure\membername)

**Result** OPCUAViewModel.OPCUAEntityReference: object instance of the desired tag. When in error will return null.



The object returned by the function is needed as a parameter in those objects inserted on screen and that belong to the toolbox.

#### Example:

```
// instance of IODataServer variable
var server = Builder.IODataServer;

// getting Tag in a folder
var tagReference = server.GetTagOPCUAEntityReference("folderName\\tagName");

// getting prototype member Tag
tagReference = server.GetTagOPCUAEntityReference("protoTagName","memberName");
```

#### 19.7.46. GetUFATag, IODataServer Method

**Syntax** UFUAModel.UFUATag GetUFATag(string tagname, [string membername = null])

**Description** Gets an UFUAModel.UFUATag object's instance of the tag indicated, possibly structure type.

Parameter	Description
string name:	name of tag to retrieve, possibly composed according to the folders in which it is contained (eg. Folder\varname)
[string memberinstance = null]	structure member name if tag is structure type. The syntax must reflect the prototype structure (eg. nomember or folder/nomember or substructure/nomember).

**Result** UFUAModel.UFUATag: object instance of the desired tag. When in error returns null.



The object returned by the function is needed as a parameter in objects belonging to the toolbox inserted on screen.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting Tag in a folder
var tag = server.GetUFATag("folderName\\tagName");
// getting prototype member Tag
var tag = server.GetUFATag("protoTagName","memberName");
```

**19.7.47. GetVarTagEntityReference, IODataServer Method**

**Syntax**      UFUAModel.TagEntityReference GetVarTagEntityReference(string tagname,  
[string membername = null])

**Description**    Gets an instance from the UFUAModel.TagEntityReference object of the tag  
indicated, if structure type.



Attention: This method only functions when the Save method has been called after adding the tag.

Parameter	Description
string tagname	name of tag to be retrieved, eventually composed according to the folder in which it is contained. (eg. Folder\varname)
[string memberinstance = null]	eventual structure member name, if the tag is structure type. The syntax must mirror the prototype structure (eg. membername or folder\membername or substructure\membername)

**Result**      UFUAModel.TagEntityReference: object instance of the desired tag. When in error will return null.



The object returned by the function is needed as a parameter in datalogger column objects.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;

// getting Tag in a folder
var tagReference = server.GetVarTagEntityReference("folderName\\tagName");

// getting prototype member Tag
tagReference = server.GetVarTagEntityReference("protoTagName","memberName");
```

**Syntax**      UFUAModel.TagEntityReference  
**(Overloaded)** GetVarTagEntityReference(UFUAModel.UFUATag tag)

**Description** Obtains an instance of the UFUAModel.TagEntityReference object of the specified tag, possibly structure type.

Parameter	Description
UFUAModel.UFUATag tag:	UFUAModel.UFUATag object type that represents the tag. An object of this type can be obtained by calling the GetTag, or as a result of an Addtag. This method also functions for tags recently added in the server.

**Result** UFUAModel.TagEntityReference: instance object of the desired tag. When in error returns null.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;

// getting the tag folder
var tagFolder server.GetFolder("tagFolder");

// getting Tag
var tag = server.GetTag("tagName", tagFolder);

// getting Tag
var tagReference = server.GetVarTagEntityReference(tag);

// getting Proto Tag
tag = server.GetUFATag("tagName", "memberName");

// getting Proto Tag
tagReference = server.GetVarTagEntityReference(tag);
```

#### 19.7.48. IODataServer, IODataServer Method

**Syntax** IODataServer(DocumentManager.ComponentService.IDocument parent, UFUAEditorManagerComponent editormanager)

**Description** This is the class builder. An already built class instance object that is created with the initialization procedure described in initialization phase (see the MoviconNextBuilder.ProjectBuilder project instance manager)

Parameter	Description
-	-

**Result** -

**Example:**

#### 19.7.49. RemoveAlarmFromTag, IODataServer Method

**Syntax**      `bool RemoveAlarmFromTag(UFUAModel.UFUATag tag, UFUAModel.UFUAAAlarmDefinition alarm)`

**Description**    Removes the assignment of a tag to an alarm.

Parameter	Description
UFUAModel.UFUATag tag	Object instance that represents the tag from which to remove the assignment to the alarm.
UFUAModel.UFUAAAlarmDefinition alarm	Object instance representing the alarm from which to remove the assignment.

**Result**      Boolean

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// getting the tag object
var tag server.GetTag("tagName");
// getting the alarm object
var alarm = server.GetAlarmDefinition("NewAlarm", server.GetAlarmSource("NewAlarmSource",
server.GetAlarmArea("NewAlarmArea")));
// remove alarm from tag
server.RemoveAlarmFromTag(tag, alarm);
// saving IOserver configuration change
server.Save();
```

#### 19.7.50. RemoveChannel, IODataServer Method.htm

**Syntax**      `bool RemoveChannel(string assemblyname, string channelname, [bool removestations = False])`

**Description**    Removes the channel from the indicated driver.

Parameter	Description
String assemblyname:	name of the driver's dll
String channelname	Channel Name
Bool removestations	Forces the removal of stations, otherwise, if the channel has stations, the removal

	will fail
--	-----------

**Result** bool: outcome of the operation, true if ok.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// remove a ModbusTcp driver channel
bool retVal = server.RemoveChannel("ModbusTCP.dll","Chan01");
// saving IOServer configuration change
server.SaveDriver("ModbusTCP.dll");
```

#### 19.7.51. RemoveDriver, IODataServer Method.htm

**Syntax** bool RemoveDriver(string assemblyname, [bool force = False])

**Description** Allows you to remove the indicated idriver.

Parameter	Description
String assemblyname	name of the driver's dll
String channelname	Channel Name
Bool force	Forces the removal of the driver, even in the presence of tags connected to it

**Result** bool: outcome of the operation, true if ok.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// remove the ModbusTcp driver
bool retVal = server.RemoveDriver("ModbusTCP.dll");
// saving IOServer configuration change
server.Save();
```

#### 19.7.52. RemoveStation, IODataServer Method.htm

**Syntax** bool RemoveStation(string assemblyname, string channelname, string stationname)

**Description** Removes the station from the indicated driver channel.

Parameter	Description
String assemblyname	name of the driver's dll
String channelname	Channel Name
String stationname	Station Name

**Result** bool: outcome of the operation, true if ok.

**Example:**

```
// instance of IODataServer variable
var server = Builder.IODataServer;
// remove a ModbusTcp driver channel
bool retVal = server.RemoveStation("ModbusTCP.dll","Chan01","Stat0");
// saving IOserver configuration change
server.SaveDriver("ModbusTCP.dll");
```

### 19.7.53. Save, IODataServer Method

**Syntax** Void Save()

**Description** Saves all the changes pending in the IODataServer server.

Parameter	Description
-	-

**Result** -

**Example:**

## 19.8. Parameters

### 19.8.1. AddParameterFile, Parameter Method

**Syntax** ScreenParametersSettings.Documents.ScreenParametersDocument  
AddParameterFile(string name)

**Description** Adds a parameter file with the specified name.

Parameter	Description
string name	name of the parameter file to be added

**Result** ScreenParametersSettings.Documents.ScreenParametersDocument (instance of an object representing the created parameter file. Returns null if in error)

**Example:**

```
// instance of Parameters object
var paramCollection = Builder.ParametersContainer;
var parmFile = paramCollection.AddParameterFile("NewParamFile")
// saving Parameters change
paramCollection.Save();
```

### 19.8.2. DeleteParameterFile, Parameter Method

**Syntax** ScreenParametersSettings.Documents.ScreenParametersDocument  
GetParameterFile(string name)

**Description** Gets the instance of an object representing the indicated parameter files.

Parameter	Description
string name	Name of parameter file desired.

**Result** Boolean

**Example:**

```
// instance of Parameters object
var paramCollection = Builder.ParametersContainer;
// deleting parameter file
paramCollection.DeleteParameterFile("NewParamFile")
// saving Parameters change
paramCollection.Save();
```

### 19.8.3. GetParameterFile, Parameter Method

**Syntax** System.Uri GetParameterUri(string name)



**Description** Gets the indicated parameter file's Uri.

Parameter	Description
string name	name of desired parameter file

**Result** Parameter file's Uri. Returns null when in error.

**Example:**

```
// instance of Parameters object
var paramCollection = Builder.ParametersContainer;
// getting parameter file
var parameters = paramCollection.GetParameterFile("NewParamFile")
```

#### 19.8.4. GetParameterUri, Parameter Method

**Syntax** bool DeleteParameterFile(string name)

**Description** Deletes the indicated parameter file.

Parameter	Description
string name	Name of parameter file to delete

**Result** ScreenParametersSettings.Documents.ScreenParametersDocument (instance of an object representing the parameter file. Returns null when in error).

**Example:**

```
// instance of Parameters object
var paramCollection = Builder.ParametersContainer;
// getting parameter file
var parameters = paramCollection.GetParameterFile("NewParamFile")
```

#### 19.8.5. Parameters, Parameter Method

**Syntax** Parameters(DocumentManager.ComponentService.IDocument parent, UFProjectManagerComponent editormanager)

**Description** This is the class builder. A class object instance which is already built and created with the initialization procedure during the initialization phase (see MoviconNextBuilder.ProjectBuilder project instance manager)

Parameter	Description
-	-

**Result** -

**Example:**

#### 19.8.6. Save, Parameter Method

**Syntax** Void Save()

**Description** Saves all the changes pending in the Parameters

Parameter	Description
-	-

**Result** -

**Example:**

# 20. Technical Specifications

## 20.1. Requirements

### Setup Requirements

In order to install Movicon.NExT the following prerequisites need to be met:

- Microsoft .NET Framework 4.8
- DirectX 9 (or later)
- Reactive Extension v1.0.2856.0
- Microsoft SQL Server Express 2014 (or later)

If the above requisites are not met, the setup will download the necessary packets from the internet and start their installation.

In this case, the setup will download and install the SQL Server Express 2019 version.



The tables below are purely indicative. As it is impossible to know the details of applications created with Movicon.NExT in advance, it is the design engineer's sole responsibility to evaluate and use a hardware that is most suitable to the characteristics of the project to be created.

### Editor Requirements

The table below reports the software and hardware requirements for developing Movicon.NExT projects using the editor.

Software Requirements	Hardware Requirements
<b>Minimal O.S.</b> Windows 8.1 Windows Server 2012 Windows Server 2012 R2	<b>Minimal</b> CPU Intel® Core™ i5-5xxx (5th generation) RAM: 4 GB HD: Hard Disk with 4 GB of free space Graphic Card: integrated
<b>Recommended O.S.</b> Windows 10 (v. 1607) Windows Server 2019 (v. 1803) Windows Server 2016*	<b>Recommended</b> CPU Intel® Core™ i5-7xxx (7th generation) RAM: 8 GB HD: Solid State Disk with 4 GB of free space Graphic Card: dedicated

### Server Requirements (OS Windows)

The table below reports the software and hardware requirements for the Server side in Movicon.NExT projects.

Software Requirements	Hardware Requirements
<b>Minimal O.S.</b> Windows 8.1 Windows 10 IOT Enterprise Windows Server 2012 Windows Server 2012 R2	CPU Intel® Core™ i3-5xxx (5th generation) RAM: 4 GB HD: Solid State Disk with 4 GB of free space Graphic Card: Integrated
<b>Recommended O.S.</b> Windows 10 (v. 1607) Windows Server 2019 (v. 1803) Windows Server 2016*	<b>Recommended</b> CPU Intel® Core™ i5-5xxx (5th generation) RAM: 4 GB, HD: Solid State Disk with 4 GB of free space Graphic Card: integrated

## Server Requirements (OS Linux)

The distributions support on Linux are exclusively those compatible with .NET Core such as: Ubuntu (Desktop and Server versions), Debian, Raspbian.

## Client Only Requirements

The table below reports the software and hardware requirements for Client side in Movicon.NEXt projects.

Software Requirements	Hardware Requirements
<b>Minimal O.S.</b> Windows 8.1 Windows 10 IOT Enterprise Windows Server 2012 Windows Server 2012 R2	<b>Minimal</b> CPU Intel® Core™ i3-7xxx (7th generation) RAM: 4 GB HD: Solid State Disk with 4 GB of free space Graphic Card: Integrated
<b>Recommended O.S.</b> Windows 10 (v. 1607) Windows Server 2019 (v. 1803) Windows Server 2016*	<b>Recommended</b> CPU Intel® Core™ i5-7xxx (7th generation) RAM: 8 GB, HD: Solid State Disk with 4 GB of free space Graphic Card: dedicated

## Client+Server Requisites

Server side software requirements (O.S.)	Hardware Requirements
<b>Minimum</b>  Windows 8.1 Windows 10 IOT Enterprise Windows Server 2012	<b>Minimum</b> CPU Intel® Celeron™ J1900 RAM: 4 GB HD: Solid State Disk with 4 GB of free space Graphic Card: Integrated

Windows Server 2012 R2	
<b>Recommended</b> Windows 10 (v. 1607) Windows Server 2016 * Windows Server 2019 (v. 1803)	<b>Recommended</b> CPU Intel® Core™ i5-7xxx (7th generation) RAM: 8 GB, HD: Solid State Disk with 4 GB of free space Graphic Card: dedicated

## Web Client Requisites (App)

The table below reports the minimum software requirements and recommendations to publish Movicon.NExT projects for Web Client (WPF) access.

Server side Software requirements, Smartphone/Tablet Software Requirements.

Software Requirements	Hardware Requirements
<b>Minimal O.S.</b> Windows 8.1 Windows Server 2012 R2 Windows Server 2012	<b>Minimal O.S.</b> Android 4.1 and later iOS 9.3 and later Windows 10 Mobile (build 1607 and later)
<b>Recommended O.S.</b> Windows 10 Windows Server 2019 (v. 1803) Windows Server 2016*	<b>Recommended O.S.</b> Android 7 and later iOS 10.3 and later Windows 10 Mobile (build 1709 and later)

## Web Client Requisites- (WPF/HTML5)

The table below reports the minimum software requirements and recommendations to publish Movicon.NExT projects for Web Client (HTML5) access.

Server side software requirements, Client side software requirements.

Software Requirements	Hardware Requirements
<b>Minimal O.S.</b> Windows 8.1 Windows Server 2012 R2 Windows Server 2012 Windows 7 SP1 <b>Other necessary software:</b> Micorsoft IIS (Internet Information Service)	The Web technology is baed on the HTML5 standard and therefore is supported by any HTML5 compatible browser with any operating system, for example, such as: <ul style="list-style-type: none"> <li>Desktop: Windows, Apple iOS, Linux, etc.</li> <li>Mobile: Windows Phone, Apple iOS, Android, etc.</li> </ul>
<b>Recommended O.S.</b> Windows 10 Windows Server 2019 (v. 1803) Windows Server 2016* <b>Other necessary software:</b> Micorsoft IIS (Internet Information Service)	

(\*) Validation of hardware requirements, in evaluation phase.

## WebHMI Client Requirements (SVG/HTML5)

The table below reports the minimum software requirements and recommendations for deploying Web-enabled projects via the WebClient (SVG/HTML5)

Server side software requirements	Browser side software requirements
<b>Minimum O.S.</b> Windows 8.1 Windows Server 2012 Windows Server 2012 R2	The Web technology is based on the HTML5 standard and is therefore supported by any HTML5 compatible browser.
<b>Recommended O.S.</b> Windows 10 Windows Server 2016* Windows Server 2019 (v.1803)	

## WebHMI Runtime Requirements

The table below reports the minimum and recommended hardware/software requisites for deploying WebHMI projects as stand-alone applications in HMI.

Hardware Requirements	Software Requirements
<b>Minimum</b> The recommended hardware on which to run WebHMI Runtime depends on many factors such as the number of tags, pages and alarms managed etc. The recommended minimum is a Raspberry Pi 4 for a project with 300 tags connected with Modbus TCP, 15 screens, 1 historian change with 30 historically logs, 50 alarms.	<b>O.S.</b> Windows 10 Windows 10 IOT Enterprise Linux compatible .NET Core (Raspbian, Ubuntu, Debian) <b>Browser</b> The Web technology is based on the HTML5 standard and is therefore supported by any HTML5 compatible browser.

(\*) The "Essentials" version of the Windows OS Server does not allow 'Typical' installation during which the local "NExT\_IO\_Server" user is created. In order to conclude the setup, you will need to select the Custom installation and indicated a domain user as user.

## 20.2. General Specifications

### 20.2.1. Users and Groups

Movicon.NExT manages user security according to strict standards using ASP.NET. Memberships. The system is therefore independent from the authentication type being used which for default is based on a SQL Server repository with the relative Microsoft authentication Provider. However other providers can be used for managing authentication (eg. biometric systems).

The project's Users management is used to define users with a series of parameters in addition to hierarchical level of privileges and access areas.

### Functionality

- User management based on ASP.NET Membership provider, with maximum security and independent from authentication provider
- The authentication provider is based on the Microsoft SQL Server repository, If not specified otherwise.
- User rights can be edited and assigned in both edit and runtime modes (settable limit of users that can be inserted in runtime)
- Supports Groups and Users management
- Electronic Signature Management
- Supports the CFR21 Part 11 and GAMP5 specifications
- Users edited in runtime uses the same list of users defined in edit mode.
- User Authentication in Web Client HTML5 architecture as well.
- Option to share Windows Operating System users (local or network domain users) for authentication
- Option to protect the project with passwords and encryption (depends on the user management).
- Option to request authentication for Operating System operations: "ALT+TAB",..."CTRL+ALT+CANC".

### Limitations

System	Limitations
Users	Max. 9999 Users per project
Groups	Max. 512 User Groups
Levels	Max. 999 hierarchical levels and 31 access areas for each individual user.

## 20.2.2. Text and String Resource

The String table and the files where the text used in the project is centralized.

### Functionality

- Supports dynamic multilanguages
- Supports Unicode
- Automatic text translation (requires internet connection)
- Option to filter inserted strings
- Import-export strings to csv files
- Text contained in the String Table can be copied to the clipboard and pasted in Microsoft Excel, and viceversa

### Limitations

Functionality	Details
Columns	Max 64 Languages (text columns)
Strings	Max 32,000 Strings for each language (records)

Characters	Max 1024 characters for each string

### 20.2.3. Windows Services

Movicon.NExT supports the Windows Services as I/O Server and for running Script logic.

#### Functionality

- I/O Data Server Service support
- Scripts Service support
- The service can be set with user credentials predefined in system or by indicating a specific user.
- When using users defined in the operating system, the service can be protected from closing by those users who are not system administrators by hooking on to the users' set rules.
- This service, as such, results already active before logging on to the operating system.

#### Limitations

Feature	Limitations
Services	As a Windows service it inherits the features and limitations of these services.

### 20.2.4. Logic

The Sequential Logic Graphics is a sequential logic editor which is programmable by inserting logic gates and associating project Tags to them.

#### Functionality

- Graphically programmable sequential logic execution
- Logic can be executed on command or event
- Logic can be also executed as an O.S. Service
- Script Blocks refer to the javascript language and fully support ECMAScript 5.1

#### Limitations

Functionality	Limitations
Logic	Only Tags and NOT numerical constants can be associated to the Input gates.
	There are no Expression fields therefore Tag bits or Array elements cannot be pointed to.
	String variable types cannot be used.
	The gates have 2 or 4 inputs and cannot be configured. However it is possible to connect several Input gates to one individual gate input, for example Or.



	Font editing is not supported therefore the text font displayed in the logic gates and comments cannot be modified.
--	---

### 20.2.5. Child Projects

The child projects render project development modular. This is done by means of associating and linking other projects to the project, both locally or in remote to create a modular and distributed project architecture.

#### Functionality

- Allows the project to be subdivided in modules: structured design.
- Simultaneous project development team work on 'child' projects to be inserted in the 'Parent' project
- The Parent project dynamically displays the resources defined in Child projects.

#### Limitations

Functionality	Limitations
	Max 32 Child Projects per Parent Project

### 20.2.6. Alarm Dispatcher

Alarm Dispatcher is an optional Movicon.NExT Server module which sends Alarm and Event notifications to user recipients by using SMS and emails.

#### Functionality

- Options to use SMS or Email for sending notifications
- Notifications can be activated or deactivated according to variable condition
- Notifications via E-mail with attachments
- Notifications to individual recipients or group recipients

#### Limitations

System	Limitations
Server	Max 2048 notification events per project
	Max 32 Users can be associated to each individual notification
	Max. size email attachments is 512 Kbytes
	Max. 10 attachments per email

### 20.2.7. Schedulers

The Scheduler is a Server module that permits commands to be managed according to scheduled time plans or calendar dates using sophisticated settings.

#### Functionality

- The Scheduler Server is an integrated server module and as a plugin it is independent from the platform and based on OPC UA in transparent mode.
- Reoccurring or cyclic expiration can be configured as needed and in runtime as well.
- Events can be managed according to a predefined calendar or a weekly plan and modified in runtime as well.
- Integrated User and Password management for display schedules and modifying settings in runtime.
- ON and OFF events can be managed with start and end dates.
- Project Tags can be associated to schedulers
- Schedulers can be displayed and set from Web Clients.
- Event recordings and scheduler server log

#### Limitations

Functionality	Limitations
Server	Max 512 Scheduler Objects per project
	Commands that can be associated to a Scheduler are a set of values in Tags.

### 20.2.8. Networking

The OPC UA connectivity and the Networking consent remote Movicon.NExT clients to connect to Servers and viceversa.

The remote clients and servers can be either Movicon.NExT application or third party OPC UA systems.

The connectivity is nevertheless based on the OPC UA data model and the Networking feature connects applications based on Movicon.NExT.

When the networking feature is enabled on the license, a server will allow a local client to connect by means of using the "netpipe" transport.

#### Transports

- NetPipe: This is the Microsoft Transport that uses shared memory. It is the most performing transport due to being the only local one. In addition it is very safe to use because it does not open any listening port on the network card and therefore no one can enter from the outside. It is used when the client and the server reside in the same PC.

- Net.Tcp : This is a Microsoft transport based on the TCP protocol to consent communication over the net. This protocol is preferential for Movicon Server and Client as an alternative to using Opc.Tcp.
- Opc.Tcp : This transport, developed by the OPC Foundation, is based on the TCP protocol. Third party OPC UA servers or devices that implement the OPC UA server inside, can provide the use of this protocol.
- HTTP and HTTPS : This transport can be used in specific configurations in public networks with or without security management.
- NoSecurityHttp : HTTP transport without security management for transmitted data.

## Limitations

Systems	Details
Net.Pipe	Max 100 concurrent sessions Max 10.000 monitored items
Net.Tcp	Max 50 concurrent sessions Max 5000 monitored items
Opc.Tcp	Max 50 concurrent sessions Max 5000 monitored items
HTTP e HTTPS	Max 25 concurrent sessions Max 2500 monitored items
NoSecurityHttp	Max 25 concurrent sessions Max 2500 monitored items

## 20.2.9. Scripts

Movicon.NExT offers the possibility to use the powerful VB.NET scripting language in your project applications on both Client and Server sides.

## Functionality

- VB.NET scripting language, multithreading with Unicode support
- Editor powerful with guide, intellisense, debugger, simulazione
- Client side run with numerous API to resources, symbols and objects.
- Supports graphical object events and methods
- Option to encapsulate code in objects and create Power Templates
- Supports Dropping Code
- Option to create methods relating to tags: this code is run within the Server context
- Support to external Assembly and User Control .NET
- Runs script code started up as Windows service
- Script can be used to reference the project's global tags and local tags by writing the name of the PlatformNext tag or specific functions.
- API and specific interfaces used for accessing functions relating to screen resources, string tables, alarms, scripts and server.

## Limitations

System	Details
Win64	Max 512 Script objects per project, of which t4 can be run in runtime simultaneously (multithreading).
Win64	Instances from the same script are not supported simultaneously
Web Client	Not available (run on server).

## 20.3. Web Client Specifications

### 20.3.1. Web Client

Platform NExT supports the web architecture to allow remote users to connect to Data Servers using the HTML5 technology.

Screens can be created to deploy web pages using the integrated application module that uses the Microsoft IIS Web Server and a ASPX application module, which connects to the Data Server via OPC UA, to deploy screen pages using HTML5.

The deploy feature creates HTML5 pages and manages connections to the Data Serve independently from the fact that the screens are displayed in the Movicon.NExT Client as well.

The mechanism to deploy and connect data is totally automatic and transparent. It offers the advantage of being able access plant system screens from any connected system using remote control with maximum interation and security.

### Functionality

- The OPC UA Web Server integrated with automatic HTML5 screen deployment
- Cross platform: possible to access from PC, tablet, smartphone (e.g. Android, iOS, Linux..) by using various browsers
- Deploy Wizard for automatically generating HTML5 Web server
- Access to all functionalities and controls on screen (with authentication option)
- Authentication and access to users defined in the project using the ASP.NET Membership provider.
- Total configurability in project designing objects, data and controls that are visible on web side
- Supports "pop-up" windows
- Complete Support to Alarms and Historical Events
- Supports Reports management
- Supports Trends and Historical Analysis
- Supports IP videocamera images managed from server
- Supports Web Sockets (full duplex) communications

## Limitations

Functionality	Limitations
Screens	<p>The Multidocument is not supported by the page structure: therefore screen used as frames are not supported on the web side.</p> <p>However "pop-up" windows are supported.</p> <p>Objects that are not supported:</p> <ul style="list-style-type: none"> <li>• Web Browser</li> <li>• Media Elements</li> <li>• Earth 3D</li> <li>• Layout Windows</li> <li>• Digital Clock with transparent background</li> <li>• Combo Box: managed as displays</li> <li>• Slider: only read</li> </ul>
Commands	<p>Commands not supported among those associated to graphical events are:</p> <ul style="list-style-type: none"> <li>• System Commands</li> <li>• User Commands</li> <li>• Alarm Commands (use commands provided on web side)</li> <li>• Report Commands (use commands provided on web side)</li> <li>• Impulsive and timed impulsive Tag Value commands</li> </ul>
3D	3D graphics not supported on Web side
Geo Scada	<p>Maps are displayed on Web side in static mode.</p> <p>Interactive geographic and cartographic maps not supported</p>
Connections	<p>Max. 10 instances simultaneously are supported on desktop operating systems via the web.</p> <p>Max. 100 instance simultaneously are supported on server type operating systems in servers via the web.</p>
Menù	The 'menu' resources are not supported.

### 20.3.2. WebHMI Functions and Limitations

Below are listed the features supported by WebHMI. Any features not shown on this list are not supported.

#### Supported Objects from the Toolbox

- Alarms: Alarm Banner, Alarm Window (without use of filters), Historian Log Viewer
- Progress Bars
- Buttons: standard and with styles
- DB Connectors: Combo-box, Grid Control
- Dialog Boxes: Check-box, Combo-Box, Spin button
- Displays
- Arc Bargraphs: all
- Gauge: circular e transparent 180°
- Meter
- Gauge Advanced

- Hot Regions
- Dynamic Images
- Lights: all
- Bargraphs: all
- Bargraph Advanced: all
- Draws: all
- Recipes: Recipe Viewer
- Schedulers: Scheduler Viewer
- Selectors: 2 position selectors, 3 position selectors (0-1-2 e 1-0-2)
- Sliders: horizontal, vertical, flat horizontal, flat vertical
- Switches: Push button switches, toggle switches
- Screen Tabs
- Texts: Animated Text, Text Box
- Trends: Chart, Trend Data Analysis, Real Time Trend, State Bar Chart, Sparkline Chart
- Viewers: Report Viewer, PDF Viewer
- Historian Data Viewers: Data Logger Viewer, Historian Viewer



The "viewer" objects of the Toolbox, such as "Alarm Window", "Event Log", "Grid", "Data Analysis Trend", etc., on the WebHMI side, can be presented with a reduced set of available commands compared to the Client Desktop version. In principle, the following may be missing:

- some commands of the object toolbar
- some command buttons
- filter functionality
- data export or print functionality
- pop-up windows (e.g. Palette window or Data Analysis Grid)

### Supported Project Resources

- Logic
- Schedulers
- Reports
- Screens
- Parameters
- Texts
- Cross Reference
- Events (excluding Events linked to System tags or local Client tags. They are supported only Events linked to I/O Data Server Tags)
- Recipes
- IO Data Server
- Users (excluding Domain users, and it is not possible to share / synchronize users between clients of different types)

### Supported Animations



Animations are only supported by the symbols from the Symbol Gallery and objects from the "Draws" group in the Toolbox.

- Background Color
- Visibility
- Blink
- Filling
- Height
- Width

- Scale
- Rotation
- Move X
- Move Y

### Supported Commands

- Screens (except for the Auto Close and Monitor ID options)
- Tag Value
- Open Pad
- Recipes (except for commands: Export/Import and Show Recipe)
- Users
- Change Language
- Reports
- Alarms
- Text to Speech
- Open Map

### Other limitations

- The .NET Core framework support is needed in stand-alone applications (WebHMI Runtime)
- Only USB hardware or software licences are supported in stand-alone applications (WebHMI Runtime), and not NET licenses.
- vb.net scripts are not supported on both server and client sides in stand-alone applications (WebHMI Runtime), but logic is available.
- Only scripts on server side run by the Script service are supported when using WebClient only (SVG/HTML5).
- Projects cannot be saved on database.
- The I/O Data Server redundancy is not supported.
- Data validation is not available for data recorded with the "Audit Trail" (CFR21),
- 3D graphics are not supported.
- Multitouch functions are not supported.
- "opc.tcp" and "http" are the only client-server communication transports available.
- Log files are saved as text files and not xml. files.
- "tiff" images are not supported.

## 20.4. Client Specifications

### 20.4.1. Screens

The screen resources are one of the essential elements of the Movicon.NExT Client module projects defined Movicon.NExT. The screens are those windows that contain the Graphical User Interface (GUI) made up of graphical drawings, symbols and objects. Process Supervision or part of it is performed by means of these screens using graphical command for animations associated to the field tags. The screen windows interface with the operator and can be activated as a whole screen or as pop-ups. Movicon also provides Embedded Screens where screens can be displayed inside other screens.

### Screen Functionalities

- Screens are based on WPF with XAML vectorial graphics.
- Top quality Graphics Library and XAML toolbox
- Integrated Vector Graphics Editor
- Top level graphical effects (blur, transparency, gradients, shadow, outer glow, etc.).

- Supports 3D graphics.
- Supports bitmap images (BMP, GIF, JPG, PNG) and multimedia formats
- Vector and 3D symbol settings in XAML format
- Centralized symbol library repository also can be managed in the Cloud
- Expandable and custom libraries
- Screen contents are independent from screen resolutions
- Screen window sizes, background color and images can be set in their properties
- Supports multi-monitor.
- Full support to Multi-touch.
- Pop-up window management (modal or synchro).
- Option to keep screen loaded in memory when not displayed.
- Native Antialiasing functionality
- Possibility to use 32 display layers for screen contents
- Page navigation with system functions (Tiles and Geo-localization) or traditional management with startup screen.
- Screen parameterization with alias.
- Engineering unit values can be used where provided in symbols and objects

### Object and Symbol graphics functionality

Each graphical element can be assigned with dynamic animation properties. The dynamic functions are:

- Blink
- Opacity
- Composed Movement
- Horizontal Movement
- Vertical Movement
- Width
- Height
- Scale
- Rotation
- Border Color
- Back Color
- Border line
- Filling
- Storyboard
- Text
- Enable
- Visual State

### Command Functionality for Toolbox Objects

Each object from the Toolbox library is predisposed to interact with the system by means of assigning tags or operation commands. The functionalities can be assigned to the object according to its type. The objects are divided into three categories:

1. Check Boxes = These objects are predisposed to manage the assigned tag values which are normally boolean type
2. Analog = These are objects that are predisposed to manage the assigned analog variable value ((integer, float, etc.)
3. Commands = These are objects predisposed to execute operation commands (typically buttons or such like)

Below is a list of operation commands that can be assigned to command, menu, accelerator and event objects:



- Set value in Tag Actions
- Command actions in Alarms
- Command actions in Screens
- Command actions in Reports
- Command actions in Recipes
- Command actions in Users
- Command actions for change language
- Command actions in operating system or in projects
- Execution and launching script actions
- Calling methods with expressions actions
- Call 3D Views Actions

## Limitations

The limitations to keep in mind when defining screens are listed in the table below:

System	Limitations
Screens	Max. 8192 screens in project. Max. 32 screens open simultaneously
Pop-up Screens	The screen background color transparency cannot be set when screen is in Modal and Frame mode. This option is however available for use in Synchro. screens.
Screen Contents	Max. 1024 vectorial elements per screen Max. 256 symbols or objects per screen Max. 256 basic elements for each symbol Max. 16 symbols contained in one symbol (nesting) Max. 16 animations simultaneously for each graphical symbol Max. 16 commands for each object
3D	Max. 1 3D model for each screen Max. 999 Elements for each 3D model Max. 32 Animations for each 3D model Max. 1 Inner Screen for each model Max. 32 Views for each model

## Expressions instead of Tags

Objects and symbols also come with the option to receive logic expressions to determine the value being managed both in Input and Output. This functionality allows Tag values, formulae and calculations to be combined in order to represent values according to how the logic has been customized to be linked to object.

The Expression method uses simplified syntax by using the same syntax used in the Microsoft Excel 2013 expressions.

System	Limitations
Expressions	Max 2 nesting levels and 8 operators.

## 20.4.2. Networking

The OPC UA connectivity and the Networking consent remote Movicon.NExT clients to connect to Servers and viceversa.

The remote clients and servers can be either Movicon.NExT application or third party OPC UA systems. The connectivity is nevertheless based on the OPC UA data model and the Networking feature connects applications based on Movicon.NExT.

When the networking feature is enabled on the license, a server will allow a local client to connect by means of using the "netpipe" transport.

### Transports

- Net.Pipe: This is the Microsoft Transport that uses shared memory. It is the most performing transport due to being the only local one. In addition it is very safe to use because it does not open any listening port on the network card and therefore no one can enter from the outside. It is used when the client and the server reside in the same PC.
- Net.Tcp : This is a Microsoft transport based on the TCP protocol to consent communication over the net. This protocol is preferential for Movicon Server and Client as an alternative to using Opc.Tcp.
- Opc.Tcp : This transport, developed by the OPC Foundation, is based on the TCP protocol. Third party OPC UA servers or devices that implement the OPC UA server inside, can provide the use of this protocol.
- HTTP and HTTPS : This transport can be used in specific configurations in public networks with or without security management.
- NoSecurityHttp : HTTP transport without security management for transmitted data.

### Limitations

Systems	Details
Net.Pipe	Max 100 concurrent sessions Max 10.000 monitored items
Net.Tcp	Max 50 concurrent sessions Max 5000 monitored items
Opc.Tcp	Max 50 concurrent sessions Max 5000 monitored items
HTTP e HTTPS	Max 25 concurrent sessions Max 2500 monitored items
NoSecurityHttp	Max 25 concurrent sessions Max 2500 monitored items

### 20.4.3. OPC UA Client

The Movicon.NExT OPC UA Client functionality permits connections to the OPC UA Server according to two different modes:

- Connecting as Client from the I/O Data Server module
- Connecting directly to display objects on the Client

The Movicon.NExT vizualization Client is natively a OPC UA Client and permits native browsing of third party OPC UA Server Items.

OPC UA is the International IEC 62541 standard

#### Functionality

- Permits connection to and browsing of any Server OPC UA Server to read or write data
- Completely supports the data Information Model according to the OPC UA standard
- Supports the DA (Data Access) specifications
- Supports the AC (Alarms and Conditions) specifications
- Supports the HA (Historical Access) specifications
- Supports differnt typies of Transports

#### Limitations

System	Limitations
Server and connected Items	see paragraph: "OPC UA Connection and Networking"

### 20.4.4. Menu

Movicon.NExT has a Menu resource which can be used for creating interactive command menus. Each menu can be composed of command items in order to create user interaction by means of using symbol or object oriented contextual menus or customized general system menus or menus for each screen window.

#### Functionality

- Execution of a command or list of commands in asscociation to each item on each individual menu
- Each Menu object can contain a set of items and dynamic items, each one with their own commands.
- Menus can be activated from any symbol or object, or from simple "hot regions" in each screen.
- Menus can be associated to specific screens to display in the active screen window's system menu bar.
- Menu Windows can be generic to all 'Main' screens

#### Limitations

System	Limitations
Menu	Max. 512 menu objects per project
	Max. 32 Items for each menu.
	Max. 16 commands for each menu item

#### 20.4.5. Shortcuts

The Shortcut resource is used to associate one or several commands or command lists to physical keyboard keys or key combinations for their immediate activation.

##### Functionality

- Activation of a command or a list of commands in association to keys or combo keys.
- Each shortcut object can contain a set of different keys, each one with a specific command.
- Shortcuts can also be associated to a specific screen for activating command keys based on which screen is active.
- All main screens can be set with a Generic set of command keys.

##### Limitations

Sytem	Limitation
Shortcuts	Max. 512 shortcuts per project.
	Max. 32 keys or combination keys for each shortcut.
	Max. 16 commands for each key or combination keys.

#### 20.4.6. Multi Touch

The multitouch operating system gestures and functionality are native to and supported in Movicon.NExT.

##### Functionality

- Independent from the hardware being used.
- Supports data scrolling, navigation and zooming according to the multitouch gestures.
- Supports object manipulation: each graphical object can be dragged, customized and zoomed as pleased within the screen. All operations are memorized according to the user who performed them.
- Supports object conditioned multitouch: commands can be activated by pressing two different objects at the same time (e.g. commands for security system)

## Limitations

Functionality	Limitations
Multitouch	Pages are scrolled using 3 fingers to avoid making mistakes by touching wrong part of screen.
	Max. 32 manipulable objects per screen
	Max. 2 objects can be used for managing double security commands.

## 20.5. Server specifications

### 20.5.1. Tags

The Tags contain dynamic information and process data. They are managed in the platform's I/O Data Server module.

The Tags can be associated to field devices using drivers or connected to several systems using OPC UA. Tags that are not connected externally are internal tags that are not connect for license purposes.

#### Functionality

- Tag names and descriptions are customizable
- Tag data types can be defined
- Structure variable types can also be defined with multiple levels (structures of structures)
- Array tag types can be defined for data types as defined below.
- String tag types can be defined.
- Tags can be made retentive (value is maintained when system shuts down).
- Two tags can be defined with the same name in two different folders: the tag's absolute name takes into account the complete path.
- Engineering Units can be assigned to tags (conversion and normalization and format rule).
- Tag Quality management for checking whether the tag has been updated correctly by the driver.
- default Initial values can be associated to tags
- Tags can be imported from the PLC database when communication driver provides this functionality.
- Generic Tag List import and export (e.g. in Excel files)
- Dynamic physical addresses of field devices can be associated directly to tags for connecting (driver, OPC or networking).
- Tags can be removed or updated automatically in various points of the project where they are being used (the refresh command in each screen is used to do this).
- Local project tags can be created. These tags will be used by objects on screen and can all be referenced using scripts.
- Aliases can be managed with parameter files
- Cross Reference can be used to see where tags are being used within the project and to trace project screen navigation commands.
- Digital and Enumerated models can be managed to display strings to substitute each number belonging to a list of numeric values.
- The "Method" model can be used to invoke methods (function) exposed by a driver.

#### Data Types

When using tags selected with Analog or Variable as model type will need to defined one of the following data types in their properties:

- Boolean
- Byte with sign (8 bit)
- Byte without sign (8 bit)
- 16 bit integer with sign (Int16)
- 16 bit integer without sign (UInt16)
- 32 bit integer with sign (Int32)
- 32 bit integer without sign (UInt32)
- 64 bit integer with sign (Int64)
- 64 bit integer without sign (UInt64)
- Float (32 bit single precision)
- Double (long) (64 bit double precision)
- String (Each byte character + 0 termination character). An Array can be created for each data type listed. Structure tags can be created when "ObjectType" model type is selected.
- When "Structure Prototypes" are presented on the card they will be listed as the Structure types available for applying to tags.

### Tag pointing

Movicon.NExT also permits tags to be addressed in the following modes:

Pointing to bits	By using the "Input Expression" field, accessible from any tag linkable object, you will be able to point to any one of the tag's bits by specifying the number using the <. BitNumber> notation where the bit number starts with the "0".
Pointing to structure members	Any one of the structure tag members can be pointed by simply dragging the structure tag member from the 'Project's Resource' to the object that it is to be linked to. Alternatively, you can select the tag from the "tag item" menu which is access from the object's adorning.
Pointing to a array element	By using the "Input Expression" field, accessible from any tag linkable object, you will be able to point to an array tag element by specifying the number using the <[ElementNumber]> notation where the element starts from the "0" index.

### Limitations

Maximum number of tags	Max. 128.000 Tags can be declared in one individual project. This is the total sum of Tag Elements or Structure Members or Array Elements Data exchanged with the field, according to license activation, is a Max. 100.000 Tags. Numbers exceeding this limit can be managed with special licenses.
Tag Names	The name of a tag must be composed of alphanumeric characters without spaces or special characters except for the ("_") character. The tag name cannot start with a special character or a number. The tag name can be a maximum of 64 characters long.
Structure Tag Size	Structure tags cannot exceed a total size of 32,767 bytes.

No. of Retentive Tags	Max 16,000 retentive tags per project. Please keep into account that the system saves the states of each retentive tag in a XML file. Each value variation is written in this file for each retentive tag. Excessive use may compromise system performance and physical memory duration.
No. of tags connected via Networking	Max. 32,000 Tags connected to other Movicon.NExT projects by means of using Networking
No. of tags with Statistics	Currently unavailable.

### 20.5.2. Alarm and Event Log

The Alarm manager is used from the I/O Data Server module. The Alarm Manager is used to define thresholds that can be associated to Tag that establish the condition type used to generate Alarm notifications according to configurable functionalities.

#### Functionality

- Alarms incorporate one or several event threshold definitions.
- Option to group Alarms by Areas and Source.
- Option to enable/disable alarms during runtime based on tag values.
- Option to insert an alarm activation delay time (delay activation filter).
- Option to set exclusive or nonexclusive thresholds for analog alarms with several thresholds.
- Option to manage alarms only when the associated tag quality is good.
- Option to associate a text and help string to alarm thresholds and String ID (multilanguages) can also be used.
- Option to assign a fixed alarm activation value or manage a dynamic value by means of using a tag.
- Option to select different alarm activation logic conditions such as "GreaterThanOrEqual", "LessThanOrEqual", "Equals", "NotEqual", etc..
- Option to associate each alarm with a severity level.
- Option to associate each alarm with a command-state tag.
- Option to associate a command list to be executed on alarm event (ON, OFF, ACK and RESET).
- Option to define whether alarm should be acknowledged and reset or a simple message.
- Millisecond data precision.
- Option to change colors for displaying alarm based on severity in the Alarm Window, for activation state and for alarm acknowledgement.
- Option to force a comment by the operator when acknowledging the alarm. The comment is linked to the alarm's severity level  $\geq 100$ . The comment will be recorded in the historical log.
- Option to send alarm text as SMS and emails using the Alarm Dispatcher server. The notification can be sent once a certain alarm conditions is verified. It can also be sent to one or several preselected recipients.
- Option to associate Script code to the alarm window for managing ACK and Reset events.
- Alarms can be defined as Templates. This means that an alarm (template) can be created and associated to several tags.
- Option to use an Alarm Statistics Manager to analyze alarm events that occurred within certain time ranges (Statistical Downtime Analysis) with custom reports.

A Status File in XML format is created in the "ALARM" folder for each alarm defined in the project in runtime. This file is used for saving alarm information when the project is closed for example the alarm's status (ON, OFF, etc.), its total ON time, its unique and transaction ID.

### Historical Event Log

- Automatic recording (excludible in the log properties) of each event linked to alarms or messages
- Records all system events
- Historical is displayed with filters such as 'by time range'.
- Print management
- Archive format in relational database (SQL Server is used for default). Open to different DB formats
- Supports Cloud (Azure)

### Limitations

The below limits are to be taken into consideration when inserting alarms:

Alarms	Limitations
Alarms	Max. 16,000 alarm objects per project
Thresholds	Max. 4 thresholds for each individual alarm Thresholds are currently fixed at 1 (trip alarm) or 4 (exclusive / non exclusive level, rate of change, deviation) for OPC UA specification conformity. The 4 thresholds can be enabled/disables individually.
Text	Max. 256 characters for alarm text strings
Severity	Max. 1000 alarm severities

Historical	Limitations
Historical Log	Max. 999 days in archive. Note that this file capacity limit depends on the limits of the databased being used as indicated below.

Memory	The Microsoft SQL Server Express database (free) is used if not specified otherwise in the settings. In this case and as indicated by the product specifications ( <a href="https://msdn.microsoft.com/library/cc645993.aspx">https://msdn.microsoft.com/library/cc645993.aspx</a> ) the limitations are: Max 10 GB for archive files, max. 1 GB of RAM used. When using the Microsoft SQL Server standard version, the limitations are: Max 64 GB of archive files, Max 524 PB, 1 GB of RAM used.  Other limitations depend on the file structure according to the producer of the Database format used and set in the project.
--------	---



### 20.5.3. Engineering Units

The Engineering units are conversion tools that can be associated to Tags. They are used to convert the Tag's raw data to into a scaled engineering unit to be displayed in the project.

The unit to be displayed can also be defined > as a tag value suffix in display objects.

#### Functionality

1. Converts the data value to an engineering unit, by performing a mathematical conversion using a ratio between the min. and max. of the raw data and the min. and max of the converted data.
2. Each single Engineering Unit can be associated to one or several Tags.
3. Texts can be associated and used as "Measure Units" that can automatically be displayed in the linked display object.
- 4.

#### Limitations

Max. number of U.I.	Max. 16,000 Engineering Units per project.
UI in Structures	The Engineering Units can be associated to structure members.
Associable Texts	Max. 64 Characters

### 4.1.1. Historian

The Historian Manager, available as an option from I/O Data Server module, offers the Historian and Data Logger functionality. Both have the task to record process data on relational database using two data recording modes respectively.

#### Historian, Functionality

- The Historian uses a model that records a series of data with a column relating to the tag name and one relating to the value. These are typically used for displaying data in graphics based on time such as Trends and DataAnalysis (Time-series data)
- Uses XPO Providers for direct DB connections
- Historical recordings are performed in Databased tables that are completely transparent to the user.
- Data can be recorded 'on time' with relative sampling time settings.
- Recordings can be set with maximum and minimum recording rates based on fast changing or unchanged tag value rates. This will avoid over cramming data files or recording nothing if value should remain stationary.
- Recording can be set to record data 'on change'.
- Option to set custom connections to the preferred database (default SQL Server).
- Option to use the Microsoft Azure technology for recording in the Cloud.
- Option to set the max. age with which to keep data in the historical
- The Historian is associated as a Template to the tags to be recorded
- Option to record only when quality is good
- Option to record data based on absolute variations, tag value percent of the previous tag value or with reference to the ranges defined in the Engineering Units.

#### Data Loggers, Functionality

- Uses a data recording model where each Tag corresponds to a DB table column, typically adapted to data groups (e.g. statistics, production, Reports).

- Uses XPO providers for direct DB connection
- Each DataLogger records data on a table with the same name of the Datalogger
- Option to select to record data "on time" with relative sampling time settings
- Option to select to record data 'on command' using an appropriate Record tag
- Option to enable recording based on tag state
- Option to reset the table (cancel all records) based on a tag's state
- Option to use a custom connection to the preferred database (default SQL Server).
- Option to use the Microsoft Azure technology for recording in the Cloud.
- Option to set the max. age with which to keep data in the historical
- Option to customize the database table and column names
- Option to add extra columns for each tag to be recorded to report: Tag Quality, Tag TimeStamp relating of the server and time of the device connected with the driver.

## Limitations

Functionality	Limitation
Historians and Data Loggers	Max 1024 Historians - Data Loggers per project
Tags for Historian	Max. 1024 Tags for each Historian
Columns for Data Logger	Max 1024 columns for DataLogger. (This number may be reduced to a max. of 341 if using aggregated tables).
Archive age	Max 9999 days can be used for archive age. The maximum age will always depend on the database engine capacity chosen (eg. SQL Express uses Max. 2GByte).
Sampling time	Not lower than 300 millisecondi. The rate in which samplings can be taken is strictly tied to the quantity of data to be recorded and the driver's sampling time. These two parameters should be taken into consideration.

Memory	<p>The Microsoft SQL Server Express database (free) is used if not specified otherwise in the settings. In this case and as indicated by the product specifications (<a href="https://msdn.microsoft.com/library/cc645993.aspx">https://msdn.microsoft.com/library/cc645993.aspx</a>) the limitations are: Max 10 GB for archive files, max. 1 GB of RAM used.</p> <p>When using the Microsoft SQL Server standard version, the limitations are: Max 64 GB of archive files, Max 524 PB, 1 GB of RAM used.</p> <p>Other limitations depend on the file structure according to the producer of the Database format used and set in the project.</p>
--------	--

### 4.1.2. Recipes

The Recipe Manager is used to create and manage data archives typically used for recording setpoint values linked to product names indexed in the archive.

The Manager permits users to set all the data values and settings and record them on DB files to be activated with the appropriated commands later when needed.

#### Functionality

- Used to build Recipe structures and define the graphical layout desired to display data.
- Records field values that relate to the recipe on a connected relational database (SQL Server is used for default)
- Supports data file format on custom DBs
- Supports data management in the Cloud with SQL Azure
- The recipe fields are divided into groups where each one can be set a task towards the field.
- Supports data transfer management from DB to Visualization and/or DB to PLC with direct data transfer management in one unique data block
- Tasks can be redefined in each member of the group they belong to
- Recipes can be called using a command or displayed through a RecipeViewer object
- Supports recipe management via Web Client
- Supports Import - Export to common text files

#### Limitations

Functionality	Limitation
No. of Recipes per project	Max. 512 recipes per project
No. of Columns per Recipe	Max. 512 Columns per Recipe

Memory	<p>The Microsoft SQL Server Express database (free) is used if not specified otherwise in the settings. In this case and as indicated by the product specifications (<a href="https://msdn.microsoft.com/library/cc645993.aspx">https://msdn.microsoft.com/library/cc645993.aspx</a>) the limitations are: Max 10 GB for archive files, max. 1 GB of RAM used.</p> <p>When using the Microsoft SQL Server standard version, the limitations are: Max 64 GB of archive files, Max 524 PB, 1 GB of RAM used.</p> <p>Other limitations depend on the file structure according to the producer of the Database format used and set in the project.</p>
--------	--

### 4.1.3. Drivers

The Movicon.NExT I/O Data Server Communication Drivers are functional modules, in the form of plug-ins, that manage communications according to the field device's data protocol to be connected to for exchanging data. The information contained in the device's memory areas are referred to by a "Physical Address" whose value is associated to the Tag.

## Functionality

- Link to PLC addresses can be managed directly in the Tag properties
- Communications are made more efficient and optimized automatically. Automatic aggregation of data in Dynamic tasks and data exchange only with those tags that are effectively in use to improve and optimize performances
- Option to customise communications, connection management and the time used to interrogate the field for tags in use or and those not in use.
- Channel concept used for speeding up communication settings for devices with the same IP address or using the same serial port.
- Automatic Import from PLC's database/program or device
- Advanced debugging and Tracing Log functions
- Statistical communication information visualization (jobs being run for which channel, station...)
- Communication quality displayed directly in each object associated to a field tag

## Performances

The communication performances depend on the quantity of data being exchanged simultaneously and the protocol being used.

For example, the table below shows various performance tests:

System	Details
PC	Win8.1, CPU i5 4 GB RAM
PLC	SIMATIC S7-314
I/O Driver	Siemens S7-TCP-IP
Exchanged Data	10,000 Tags (16 bit) simultaneously
Performances	1.15 secs refresh time

## Limitations

The limitations to be taken into consideration while defining drivers are reported in the table below. Note: consistent data aggregation and good configuration will improve performances.

System	Limitations
Win64	Max 32 drivers simultaneously
Number of Stations	Max 128 Stations for each individual driver used.
Data Structure	On Linux, only the following drivers are supported: BACnet <ul style="list-style-type: none"><li>• Allen-Bradley Rockwell - Ethernet/IP</li><li>• Modbus TCP-IP Master</li><li>• MQTT Client</li><li>• Omron - Ethernet/IP</li><li>• Siemens - Ethernet S7 TCP</li></ul>

#### 4.1.4. OPC UA Server

The Movicon.NExT I/O Data Server module is a native OPC UA server, certified according to the OPC Foundation standard.

OPC UA is defined as the international IEC 62541 standard.

##### Functionality

- Permits connection of any OPC UA client to read or write data
- Completely supports the data Information Model according to the OPC UA standard
- OPC Server Certification by the OPC Foundation test laboratories.
- Supports the DA specifications (Data Access)
- Supports the AC specifications AC (Alarms and Conditions)
- Supports the HA specifications (Historical Access)
- Supports different types of Transports

##### Limitations

System	Limitations
Server and connected Items	See paragraph: "OPC UA Connections and Networking"

#### 4.1.5. Events

The Even Manager is a Movicon.NExT resource that is used to activate the execution of one or more commands based on programmed events such as on status change, on events in Tags, or cyclic events according to scheduled times.

##### Functionality

- Possibility to enable events dynamically using tags
- Possibility to set condition in tags to determine events ; ">",">=","<","<=","<>","on change".
- Possibility to managed events based on a predefined cyclic time schedule using minutes, hours, specific days, months and years.

##### Limitations

Functionality	Limitations
Server	Max 1024 Event Object per project
Server	Max 16 Sequential commands per each Event Object

#### 4.1.6. Redundancy

The Redundancy management (optional) provides Hot Backup and Fault Tolerance for redundant mission critical station redundant networks. It permits an Active Server to perform as Primary and other active Servers to perform as Secondaries with full operativity and complete synchronization of all functional states and historicals being managed. Any connected Clients can manage eventual server switch over according to operativity mode and server load balancing purposes.

##### Functionality

- Fully automated surveillance and entry into redundant server array service management
- Fully automated data synchronization management
- Redundant communication driver management
- Historical and Datalogger synchronization management
- Active Alarms and Historicals redundancy management
- Archive and Server alignment functions
- Server names listed in hierarchical order of activation in redundancy mode.
- Option to use a number of threads dedicated to synchronizing historicals
- Redundancy tag to indicate the active server
- Option to align historicals to the time set or with a time range
- The last server that took over control can be kept active when the primary server is restored (managed only with two servers)

##### Limitations

Functionality	Limitations
Server	Max 4 servers managed simultaneously
Server	Scripts, recipes, shedulers and the alarm dispatcher are not managed in redundancy mode.
Tags	Max 100,000 Tags managed in redundancy
Tags	System tags are not managed in redundancy mode





Progea | Machine Automation Solutions

**Emerson Automation Solutions**

Progea Srl

Via D'Annunzio 295 | 41123 Modena

(MO) | Italy

+39 059451060

**[www.progea.com](http://www.progea.com)**

Copyright information:

ConnexT© and Movicon NExT© are software products and a copyright of Progea.

All rights are reserved.

The Emerson logo is a trademark and service mark of Emerson Electric Co. Progea is a mark of one of the Emerson Automation Solutions family of business units.

All other marks are property of their respective owners.