



# Movicon NExT

## 4.2 Logic

Ver.3.4.268



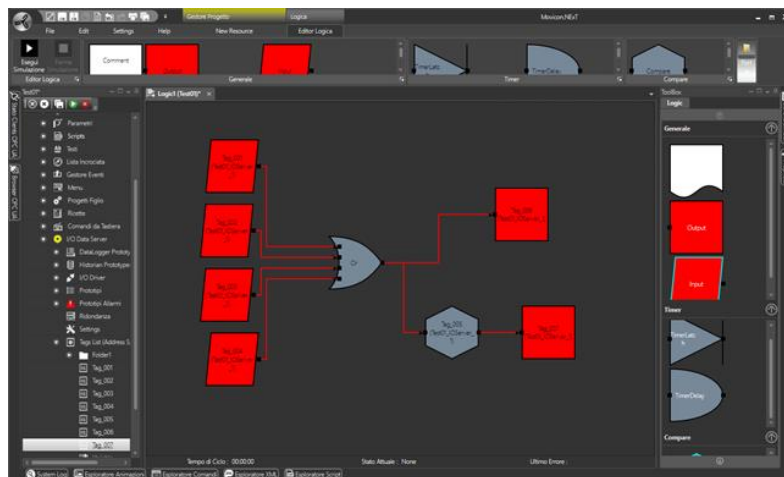
# Table of Contents

<b>1. SEQUENTIAL LOGIC .....</b>	<b>1</b>
1.1. RUNNING AND STOPPING LOGIC .....	1
1.2. LOGIC FUNCTION BLOCKS AND GATES.....	2
1.3. ASSIGNING TAGS.....	3
1.4. EDITING LOGIC .....	3
1.5. LOGIC RESOURCE PROPERTIES.....	3



# 1. Sequential 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).

## 1.1. Running and Stopping 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.

## 1.2. Logic Function Blocks and Gates

The function blocks that can be used within logic are the following:

### Comment

The "Comment" function block is only used for inserting a comment in the logic. The inserted comment does not have any influence on how the logic is executed.

### Input and Output

The Input and Output gates, as indicated by their name, are gates to which the input Tag and output process are associated. The "output" function blocks activate the output, by setting the associated Tag, when the input condition is verified as True.

### Compare

The "Compare" function block performs a comparison operation between two Tags or between a Tag and a constant value. When two Tags are associated to the function block using the appropriate properties, the comparison will be performed between the two tags. If only one Tag is associated to the function block, the comparison will be performed automatically in respect to the constant value defined in the respective property. The selectable comparison types are: Equal, Greater, Lower, GreaterandEqual, LowerandEqual. The Compare function block will activate its output when the input condition is verified (True) and the comparison result is verified.

### Timer

There are two types of Timer function blocks, "TimerDelay" and "TimerLatch". The "TimerDelay" starts countdown when the input condition is True and activates output when the time runs out. The countdown will be zeroed in the absence of the input condition and as a consequence the output will be set to zero. The "TimerLatch" activates output as soon as the input condition is True and its output will be set back to zero when the time runs out. The countdown will be zeroed in the absence of the input condition and as a consequence the output will be set back to zero.

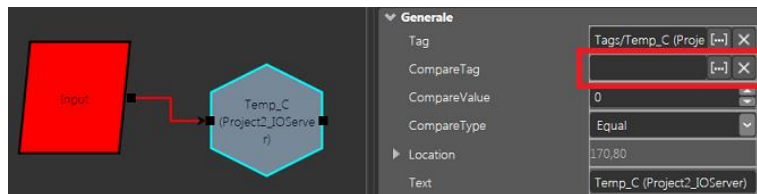
### Logic Gate

The Logic Gates are available as "OR", "AND", "NOR", "NAND", "XOR", "NOR", "XNOR" and "PID". These gates can have two or four inputs. The output of these logic gates will

be activated when the input values satisfy the conditions specified by the different logic gates.

## 1.3. 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).

## 1.4. Editing Logic

### Logic Ribbon

When opening the Logic Editor with a click, a Ribbon will show with the various logic function blocks and two commands to start and stop test runs of inserted logic.

- The **"Start Test"** command runs the logic. In this phase the values of the Input function blocks can be forced by double clicking on the actual function block to control the logic's correct flow.
- The **"Stop Test"** stops the test run. The Logic Editor will however remain active throughout the test run.

### Logic ToolBox

The toolbox containing the function blocks for inserting the logic will activate When opening the Logic Editor. The ToolBox may also have Tabs containing screen objects, but it is not possible to insert objects from the Screen's ToolBox or Symbol Gallery in logic function blocks. The same goes for function blocks from the Logic ToolBox which cannot be inserted on Screen.

## 1.5. 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.

## 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



## General

### **SessionName**

When specifying a name in this field, the "Logic" will be executed in a Server connection session that is different to the one used for the Movicon Client. If the logic runs in a different session, the connection parameters to the Servers used will be those defined in the Logic resource as described below. If the Logic is run in the same session as that of Movicon Client, the "SessionName" should then be left empty and the Server connection parameters will be the same as those used by Movicon Client which are defined in the project's "Connection Settings" properties.

### **RemoveDisabledItemAfterSecs**

This parameter is only managed if the Logic's "SessionName" property has been set. It is used for setting a delay time for removing inactive OPC UA Item subscriptions. The value used here is in seconds.

### **MaxCleanCount**

This parameter is only managed if the Logic's "SessionName" property has been set and is used for setting the maximum number of inactive OPC Items to be removed at each time interval.

### **UseAlwaysSecureConnections**

This parameter is only managed if the Logic's "SessionName" property has been set. When enabling this option only 'secure' connections to the Server will be used.

### **FastSamplingInterval**

This is only managed if the Logic's "SessionName" property has been set. This field is used for setting the update frequency of an existing variable in-use. This parameter is passed to the Server and any eventual Driver when the Logic is loaded and the variables go into use.

### **SlowSamplingInterval**

This is only managed if the Logic's "SessionName" property has been set. It is used to set the update frequency for an existing variable that is just about to go into use. This parameter is passed to the Server and any eventual Driver when the Logic is loaded.

### **DisableWhenNotUsed**

This is only managed if the Logic's "SessionName" property has been set. Sets Tags as "Inactive" when not in use.

### **PublishingInterval**

This is only managed if the Logic's "SessionName" property has been set. Tag notification time towards Server. The value is expressed in milliseconds.

