MultiTagViewer

Document Information

Software Version:	4.0.2.0
Creation Date:	30 September, 2020
Last Edit Date:	17 February, 2021
Version:	1.2

1. Scope

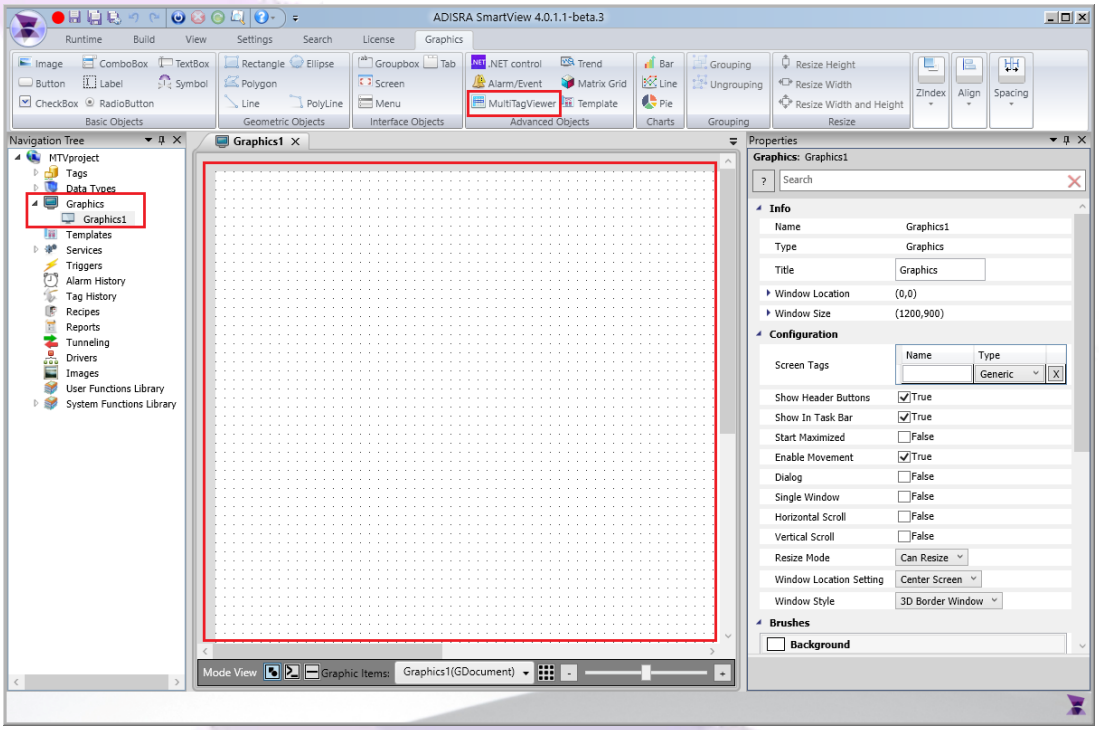
This document will present the ADISRA SmartView graphic object MultiTagViewer, its configuration options and layout during execution.

2. Summary

The MultiTagViewer (MTV) displays data from different types of tags, mostly used to display arrays of data types or simple arrays, but it can also display simple tag properties. It helps the user to create a functional table with different types of columns.

3. MTV Configuration

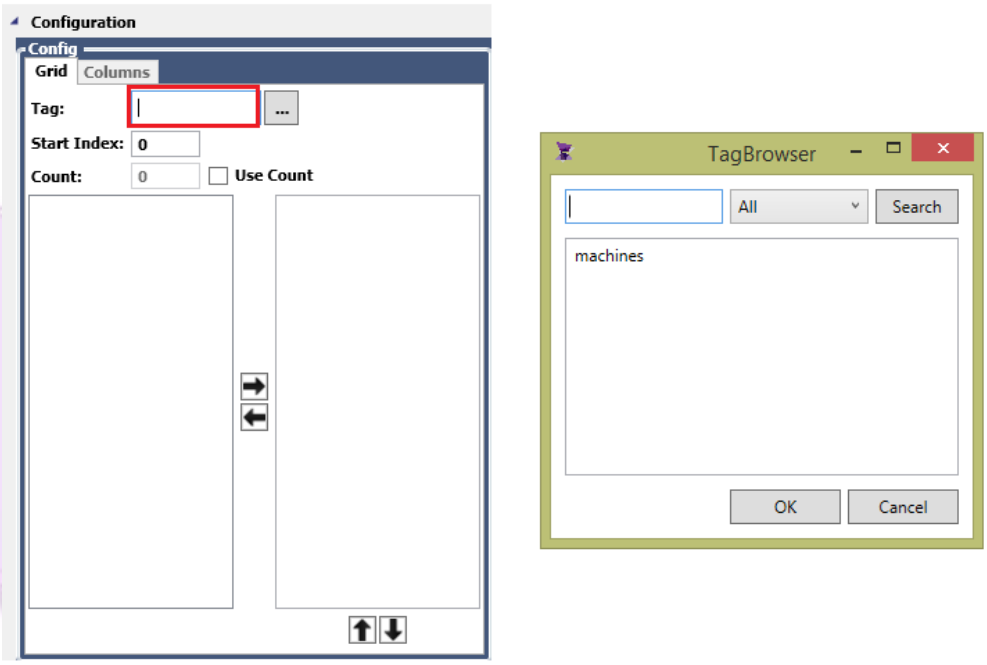
To create a new MTV object, create or open a graphic document, then select the "MultiTagViewer" object and click on the graphic area:



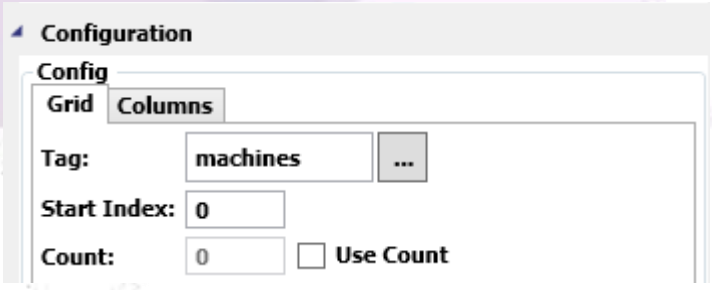
3.1. Adding Tags to the MTV

The MTV can be used to display values from a simple Tag or DataType Tag, an array of tags or an array of DataType tags. Please review the instructions below to understand how to add a tag to the object.

a. In the config section of the MTV properties, please select the configured Tag by either typing the @ sign to select the Tag or click the "…" button to open the Tag Browser window and select the Tag:



b. Start Index configures the array's initial index that will be shown on the MTV. The Count option may also be configured to determine how many items the MTV will display after Start Index in case the Tag configured is an array (i.e. It is possible to display the last 10 items of an Array Tag of size 20. To accomplish that, the Start Index must be configured with value 9 since the first position of an array is 0 and the Count must be configured with value 10):

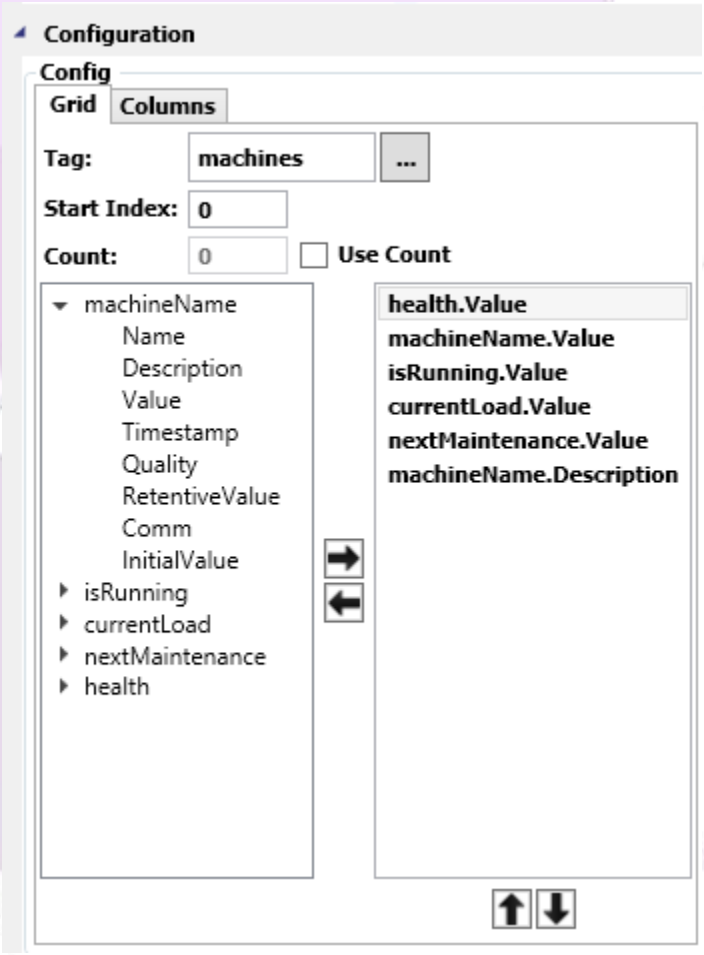


c. The left list box displays the inserted tag structure. It can be either a simple tag with properties such as name, description, quality, value or it can be a DataType with inner tags and their respective

properties. In the example below, a "health" DataType tag was added with inner tags: health, machineName, etc.

d. Use the arrows to add or remove columns to the MTV and change their order.

e. In this example, all the inner tags' values were added to be displayed as a value in columns. However, the last inner tag named machineName.Description will be used to display a button and it will have a script associated with it.



3.2. Column Types

The table below displays the column types and their usage.

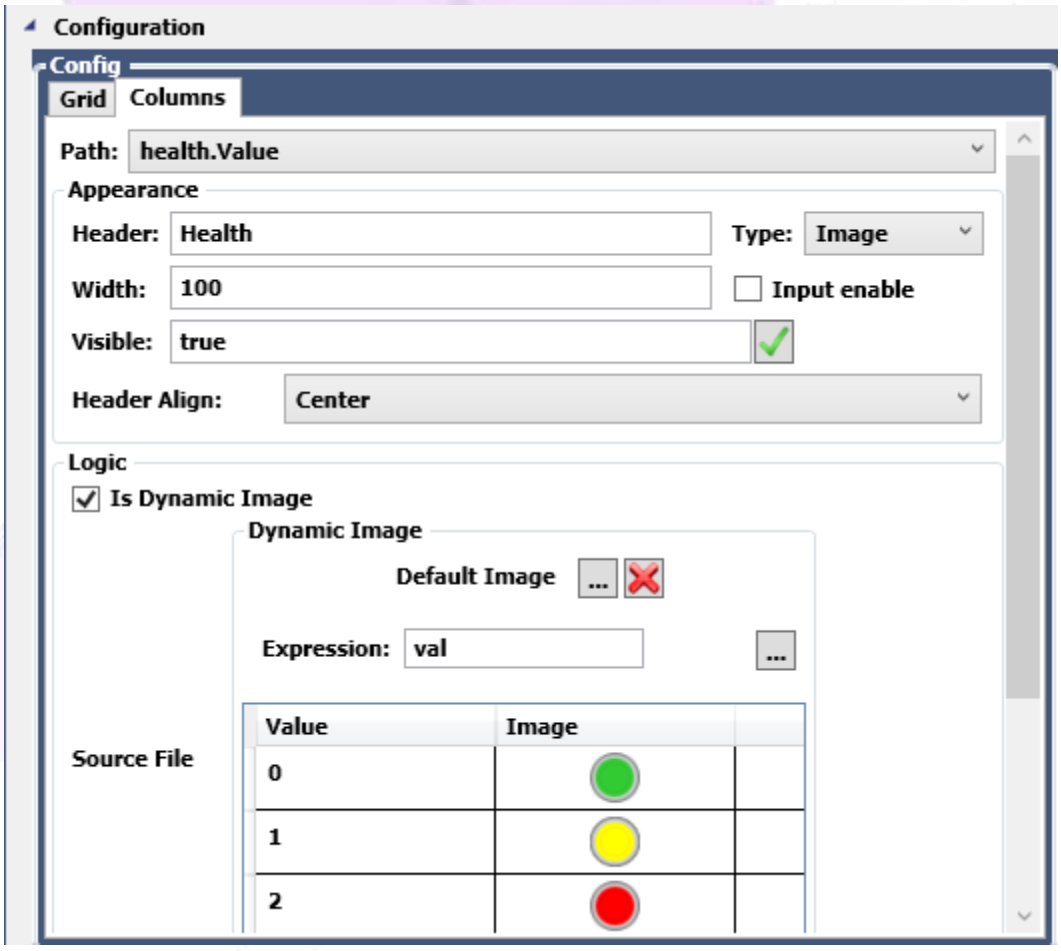
Type	Usage
Text	Can be used to display a tag's value and for user input. When displaying a value, it is possible to modify the cell value by concatenating or linking the tag's value with a unit or converting true and false to "on" and "off".
Button	Changes the grid cell into a button and allows the user to execute a C# script. The button text can also be customized for each row. The scripts receive the value and the index of the grid cell which allows the user to create powerful scripts.
Image	Instead of showing the tag value, it will display an image and it can change dynamically depending on the tag's value of that row. For example, an integer status can be converted into different images such as a closed/opened valve or a semaphore green/yellow/red circle.
CheckBox	It can represent a Boolean tag. Input can be enabled to allow the user to change the tag's value. The text next to the checkbox can be modified or hidden.
ComboBox	It can be used to list values from an array or static fields previously defined. A script can be written for a Selection Changed event allowing the user to implement flexible logics.

3.3. Column configuration options

The MultiTagViewer has multiple configuration options, some of them common to many other Graphic objects such as Size, Location, IsVisible, IsEnabled, Fonts and Brushes but others are unique.

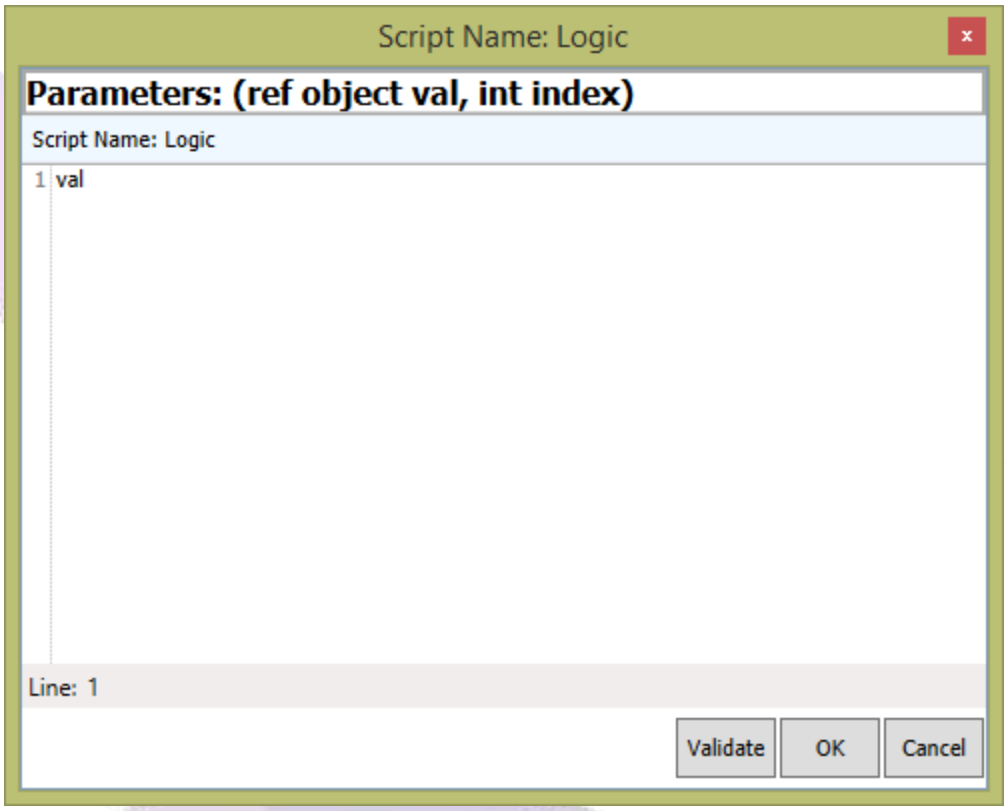
Let's take a look at the column configuration options described in the previous sub-chapter.

a. Using a dynamic image to display different images during Runtime according to the value of the tag. In the example below, 3 different images were associated to different values (0 – green circular image, 1- yellow circular image, and 2 – red circular image).



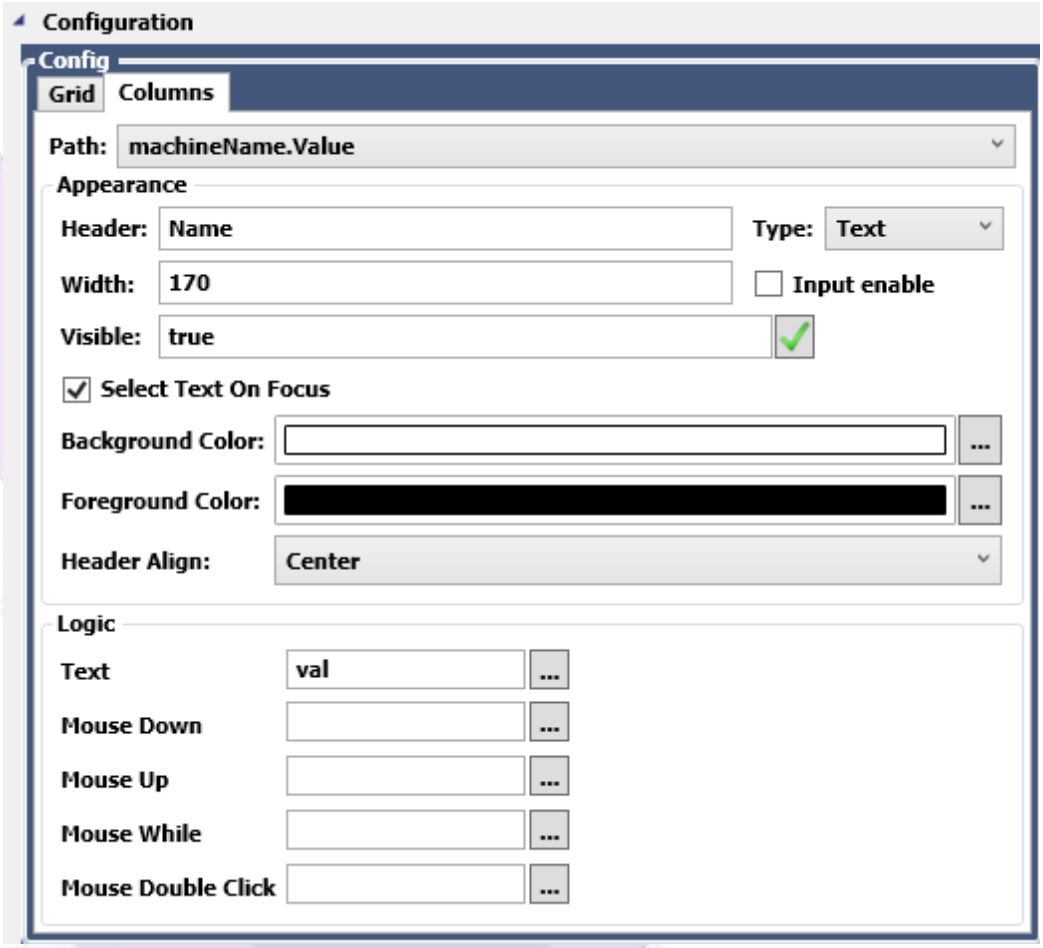
An expression is used to evaluate the value returned. In this example, it is using the current tag value (val) without further conversion. It is possible to create a C# ternary expression if necessary. Please take a

look at the expression box below to understand the parameters available.

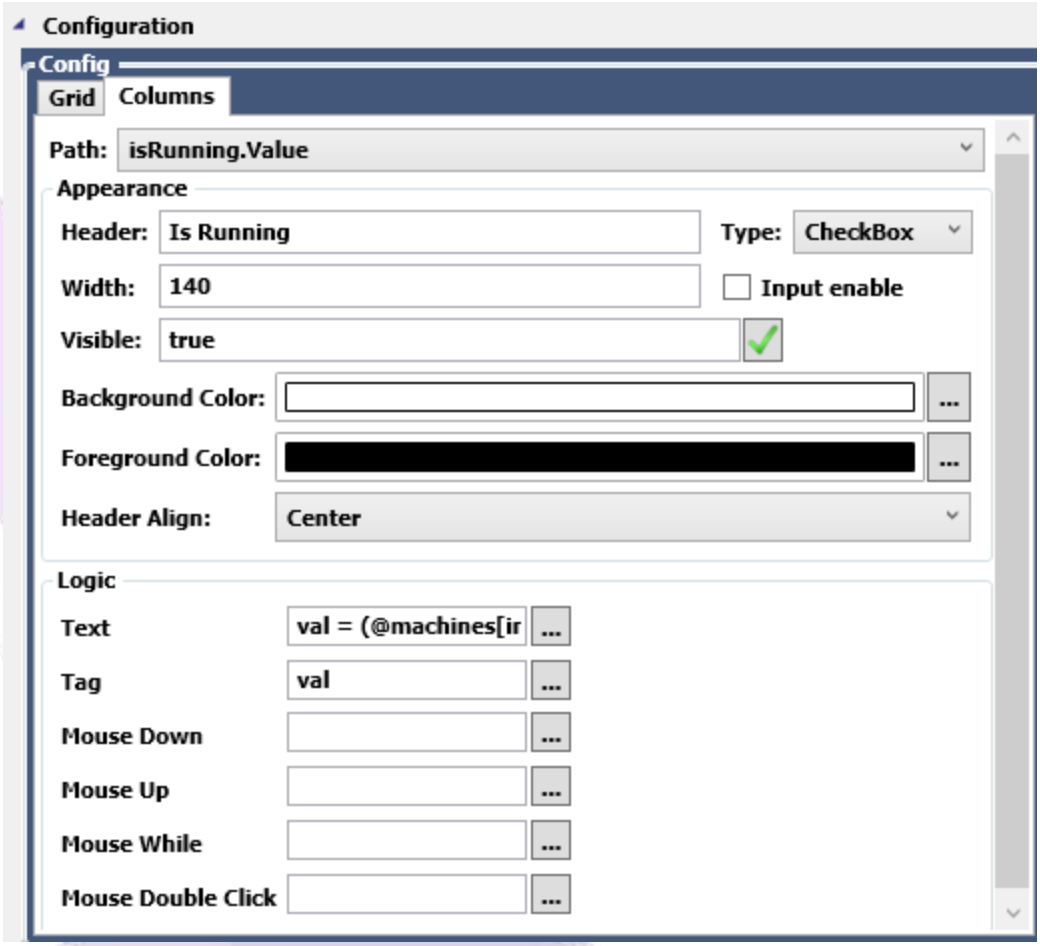


The Logic script has 2 parameters, a ref object val and an integer containing the row index. With these parameters, it is possible to access any tag from the current row and change the val value. However, in this example the val already contains the tag's value from 0 to 2 so the final expression is simply the val object.

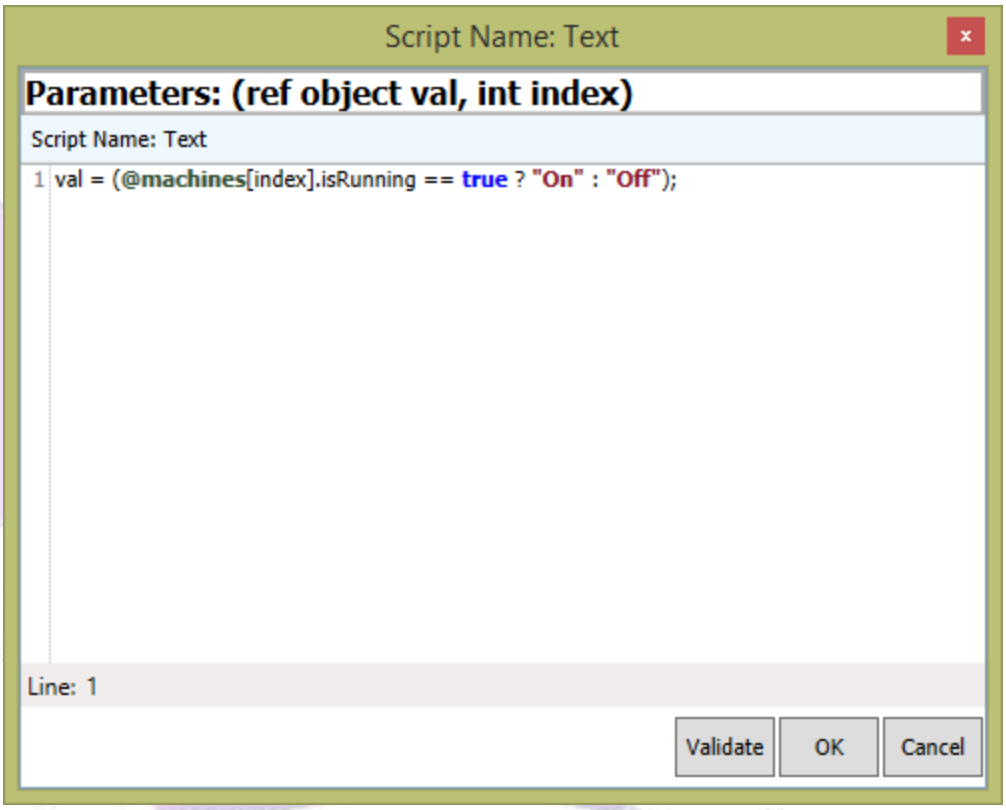
b. Displaying a simple text where the input is disabled. In this example, the tag value will be directly displayed without any changes so the expression is again the val value.



c. Displaying a checkbox to represent a Boolean and customizing the text in the expression box.

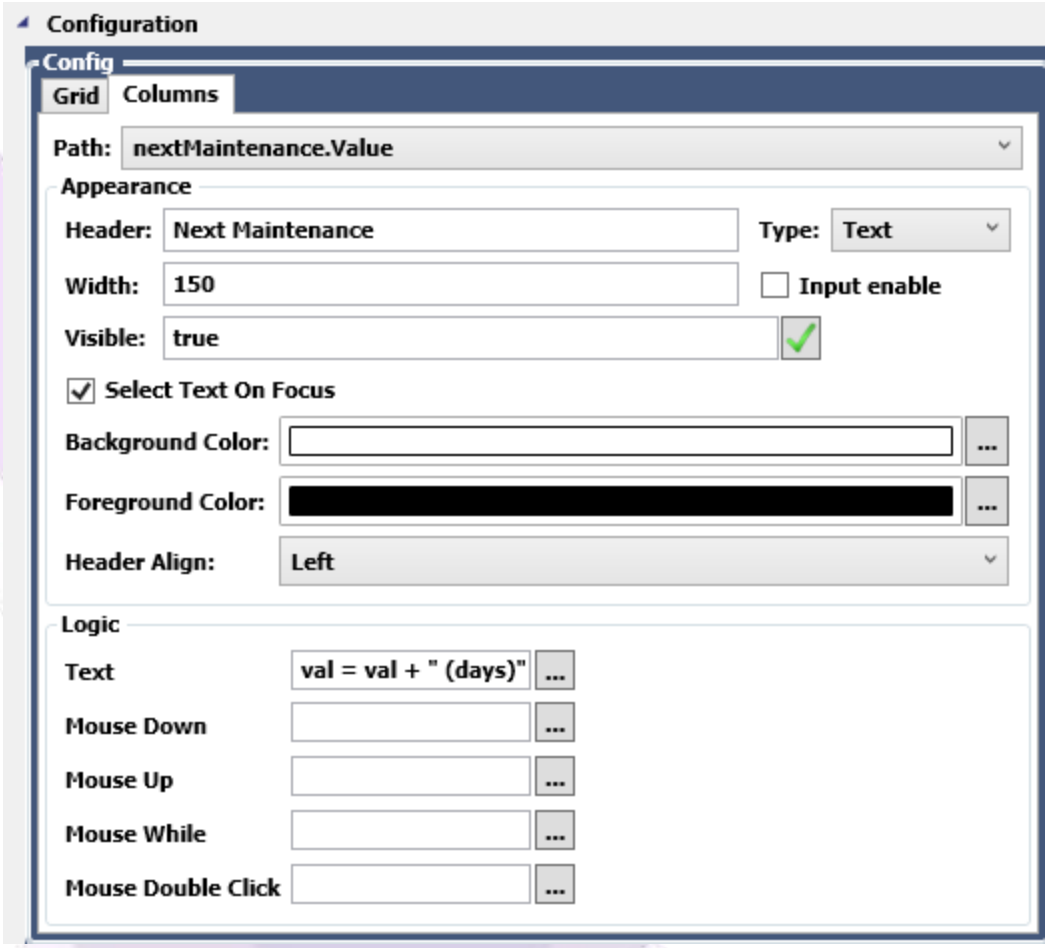


In this example, the Text is not the tag value, but it is a string "on" or "off" depending on the tag's value. Please take a look at the image below to understand how a C# ternary expression is written.

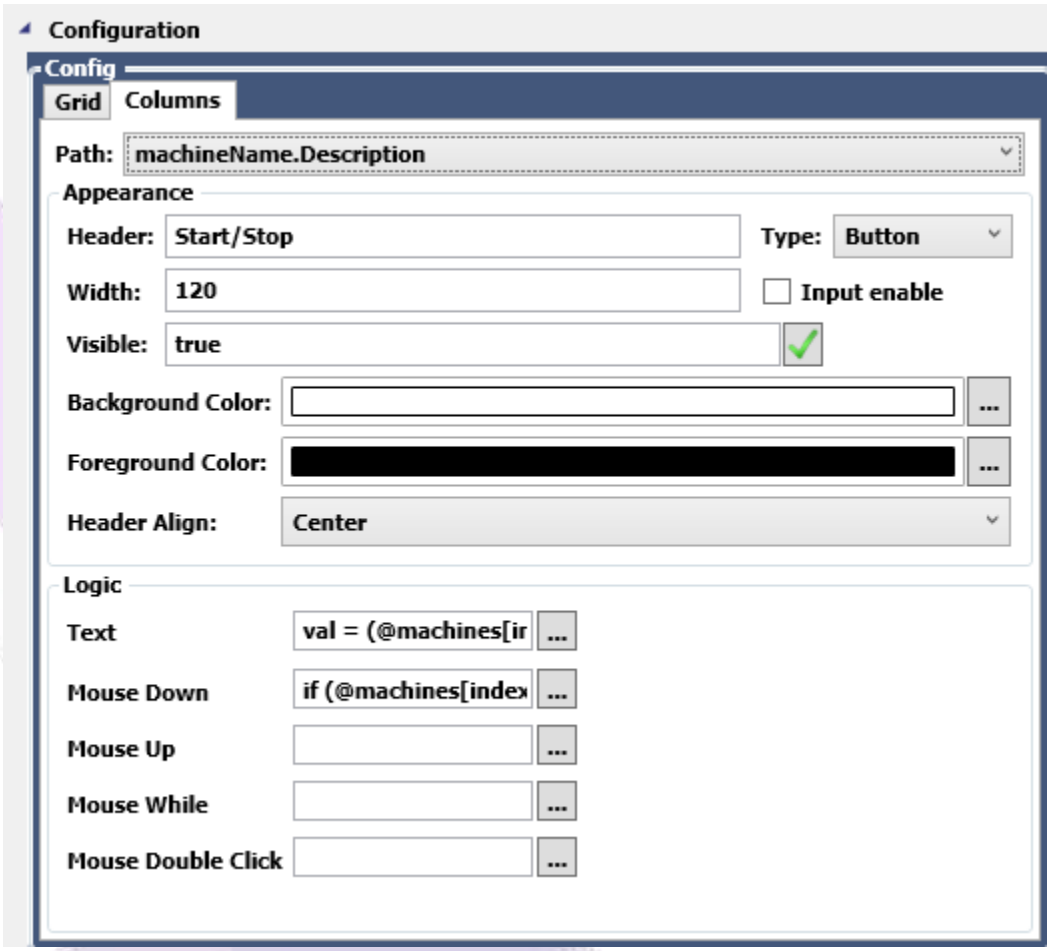


It checks for the Boolean tag of the current row using the index integer parameter. If it is true, the val value will receive "On"; otherwise, it will receive "Off". Since val is a reference parameter, it will return that new value to the grid.

d. Instead of a ternary expression, it is also possible to concatenate or link the tag's value to a string or convert the value to a different unit. In the example below, the next maintenance is concatenated to " (days)" string.

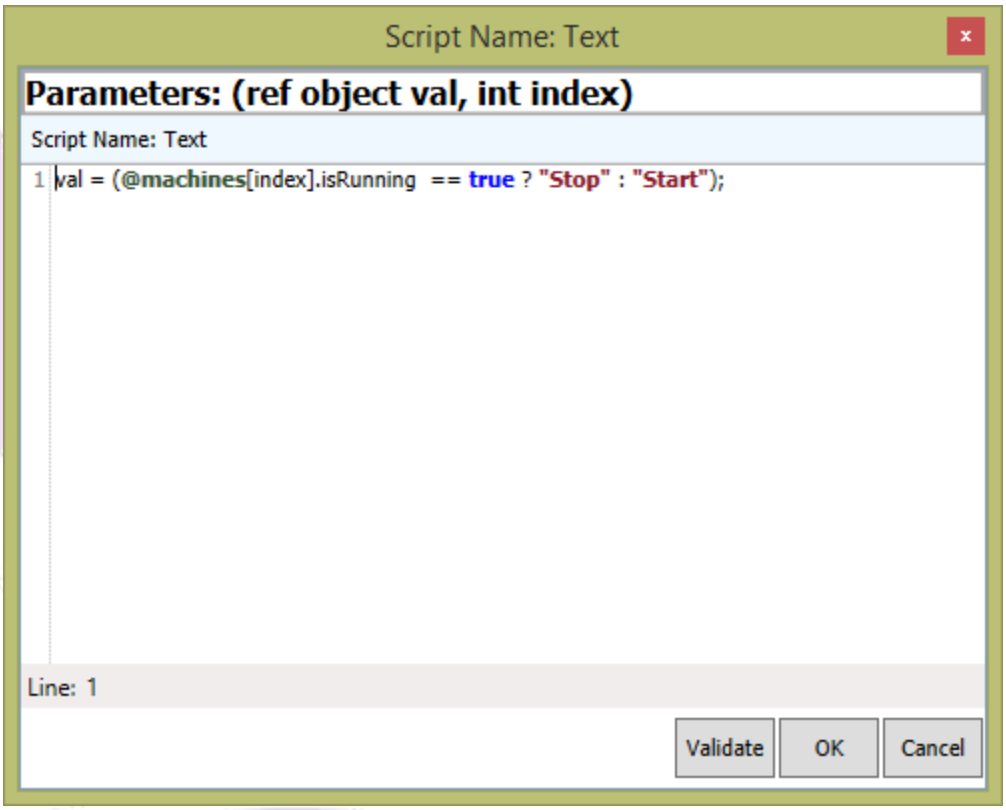


e. A column can also be changed into a button type as in the next example. If a button is needed to start or stop a machine in production, a checkbox could be used or a text but there is also the option to use a button. Within the button, we will create a script to validate a checklist before actually starting or stopping a machine. We can customize the button text using a ternary expression.



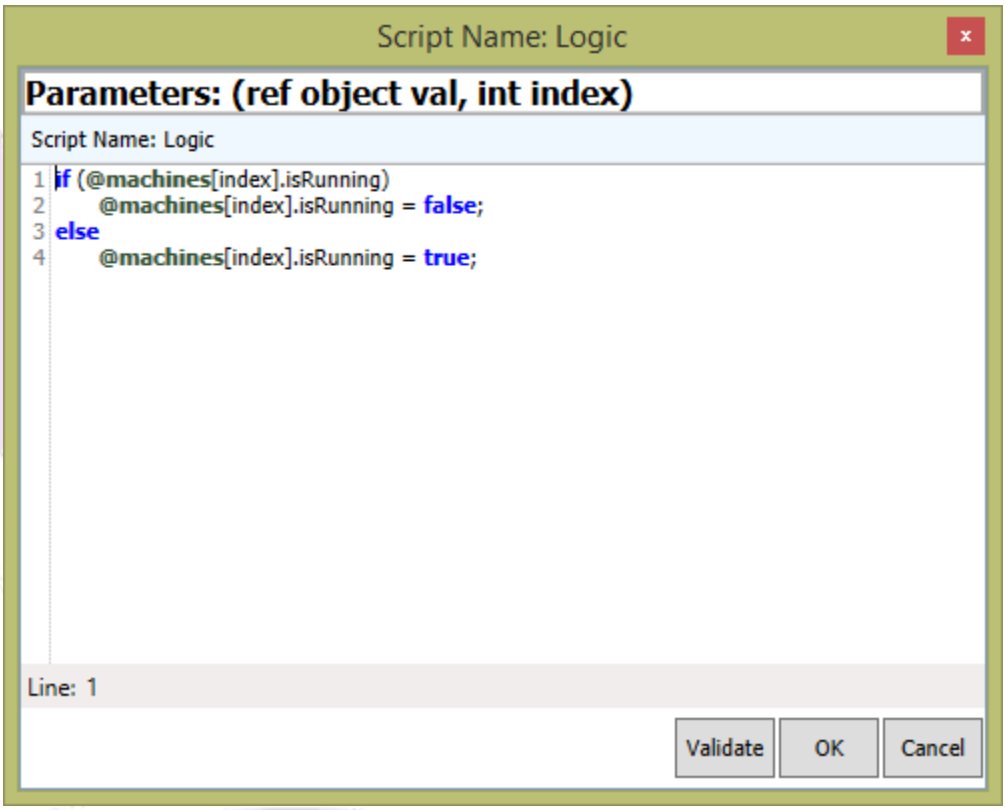
Both a ternary expression and a mouse down event have been added to the button. Let's review them separately.

Button Text Expression:



The button text expression checks if the machine is running and what text should be displayed. If the machine is running, the button text will show “Stop”. If the machine is not running, the button will show “Start”.

Mouse Down Script:



The script is not simply a C# ternary expression. It allows establishing local variables, calling user functions, using .net framework functions, and writing powerful code. In this example, we are changing the isRunning tag to true or false depending on the tag's current state. The scripts also make use of the input parameters val and index.

3.4. General Column configuration options

Header: The column name that will be displayed in the MTV.

Type: Change the column type: Text, Button, Image, CheckBox or ComboBox.

Width: The column width start size, it can be changed manually during Runtime.

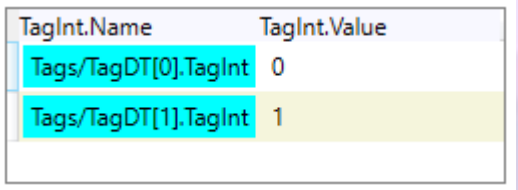
Input Enable: Configure whether the user can change the value of the selected column cell in the MTV.

Visible: Determine if the column will be visible. True to be always visible; false to be always invisible; or configure an expression to be visible or not depending of the result.

Select Text On Focus: Clicking a cell will select the text.

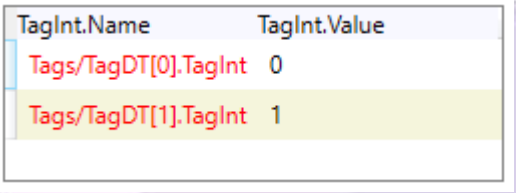
Is Dynamic Background: Determine if the column's cells background color will be dynamic or not. It operates like the brushes dynamic color.

Background Color: Configure the column's cells background color. It can be a solid color, gradient color, or an image.



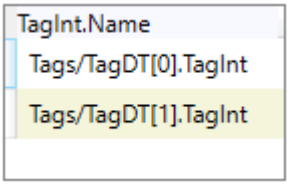
Background Color: Aqua

Foreground Color: Configure the column's cells text color. It can only be a solid color.

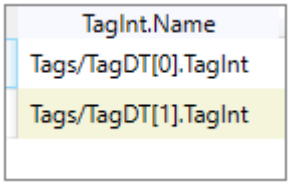


Foreground Color: Red

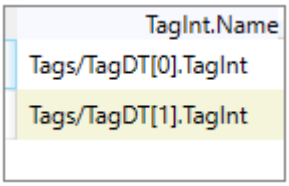
Header Align: Configure the header text alignment. It can be left, center, or right.



Left



Center



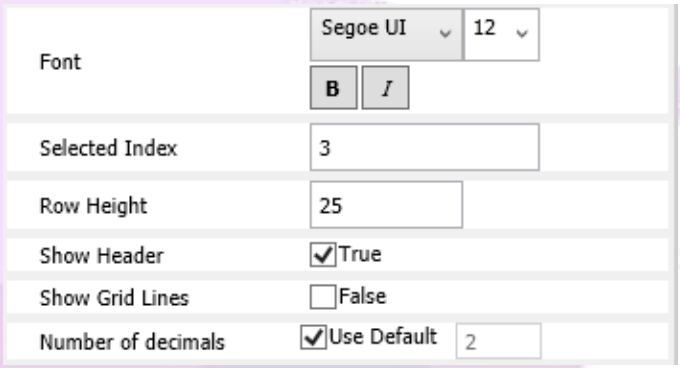
Right

Logic: The area to configure the column's logic in the MTV, different column's types will have different options.

Text: Determine the cell text. In the example, it is val, the actual TagInt name.

Scripts: Configure scripts for the Mouse Down, Mouse Up, Mouse While and Mouse Double click actions.

See the image below for more MTV object configurations:

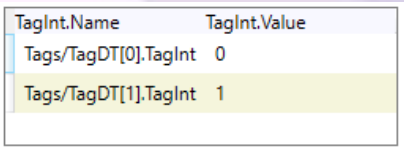


Font: Configure the font style, font size, bold, and italic for all text in the MTV.

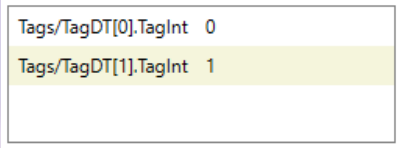
Selected Index: Configure a tag in this field to manipulate the items selected, know what ComboBox index is selected, or type in a simple number to define what index will be selected when the graphic opens.

Row Height: Determine or change the object's row height.

Show Header: Determine to hide/show the object's header.

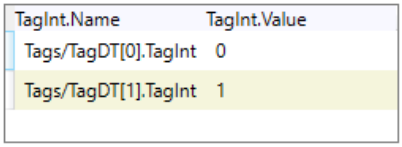


Show Header: True

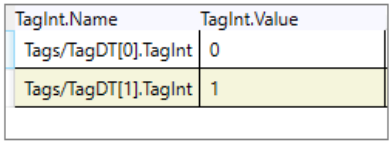


Show Header: False

Show Grid Lines: Determine to hide/show the object's grid lines.



Show Grid Lines: False



Show Grid Lines: True

Number of Decimals: Configure the number of decimals that will be shown. In this example, it is a float value.