



MultiTagViewer

Document Information

Software Version:	V4.0.3.3
Creation Date:	30 September, 2020
Last Edit Date:	9 March, 2022
Version:	1.1

Table of Contents

1. Scope.....	3
2. Summary	3
3. MTV Configuration.....	4
3.1. Adding Tags to the MTV	4
3.1.1. How to Add a Tag to the Object.....	4
3.1.2. Start Index.....	5
3.1.3. Tag Structure	5
3.1.4. Columns: Add, Remove, Change Order	6
3.1.5. Example	6
3.2. Column Types	7
3.3. Column Configuration Options	7
3.3.1. Using a Dynamic Image.....	8
3.3.2. Display Simple Text Where Input Disabled	10
3.3.3. Displaying a Checkbox Represent Boolean	11
3.3.4. Concatenate or Link the Tag's Value.....	13
3.3.5. Column Changed to a Button Type.....	14
3.4. General Column Configuration Options	16
3.4.1. Header	16
3.4.2. Type	16
3.4.3. Width	17
3.4.4. Input Enable.....	17
3.4.5. Visible.....	17
3.4.6. Select Text on Focus	17
3.4.7. Is Dynamic Background.....	17
3.4.8. Background Color.....	17
3.4.9. Foreground Color.....	17
3.4.10. Header Align.....	17
3.4.11. Logic	18
3.4.12. Text	18
3.4.13. Scripts.....	18
3.4.14. Font	19
3.4.15. Selected Index.....	19
3.4.16. Row Height.....	19
3.4.17. Cells Spacing.....	19
3.4.18. Show Header	19
3.4.19. Show Grid Lines.....	19
3.4.20. Number of Decimals	19

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.

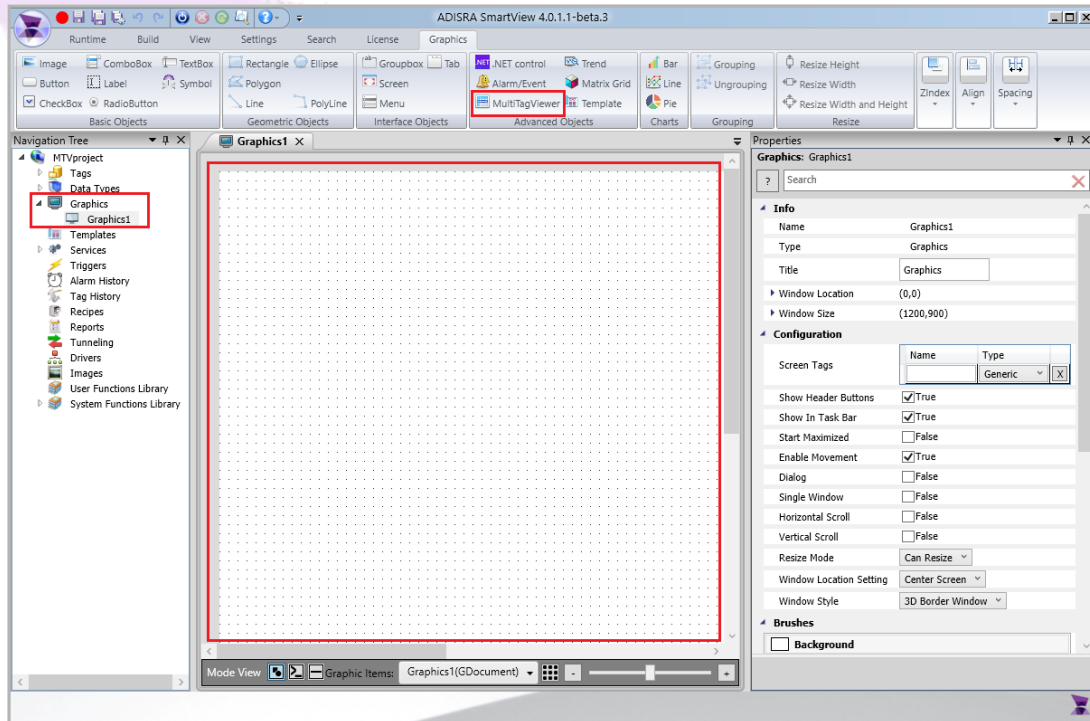
The screenshot shows a window titled "MultiTagViewer Demo" with a dark blue background. At the top, the word "Machines" is displayed. Below it is a table with the following columns: Health, Name, Is Running, Current Load, Next Maintenance, and Start/Stop. The table contains 12 rows of data for machines Machine0 through Machine11. At the bottom of the window, there are two buttons: "Reset" and "Hide/Show MTV Button".

Health	Name	Is Running	Current Load	Next Maintenance	Start/Stop
Red	Machine0	<input checked="" type="checkbox"/> On	50,00	2 (days)	Stop
Yellow	Machine1	<input checked="" type="checkbox"/> On	51,00	3 (days)	Stop
Green	Machine2	<input type="checkbox"/> Off	52,00	4 (days)	Start
Yellow	Machine3	<input checked="" type="checkbox"/> On	53,00	5 (days)	Stop
Green	Machine4	<input type="checkbox"/> Off	54,00	6 (days)	Start
Yellow	Machine5	<input checked="" type="checkbox"/> On	55,00	7 (days)	Stop
Green	Machine6	<input checked="" type="checkbox"/> On	56,00	8 (days)	Stop
Yellow	Machine7	<input type="checkbox"/> Off	57,00	9 (days)	Start
Green	Machine8	<input checked="" type="checkbox"/> On	58,00	10 (days)	Stop
Yellow	Machine9	<input type="checkbox"/> Off	59,00	11 (days)	Start
Green	Machine10	<input checked="" type="checkbox"/> On	60,00	12 (days)	Stop
Yellow	Machine11	<input type="checkbox"/> Off	61,00	13 (days)	Start

Buttons: Reset, Hide/Show MTV Button

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:

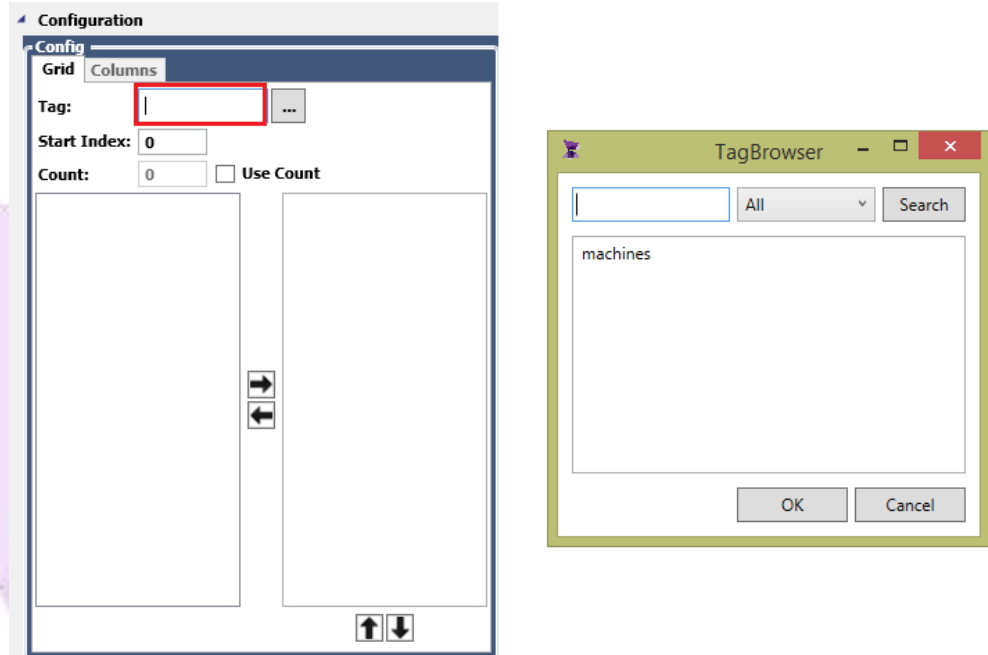


3.1. Adding Tags to the MTV

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.

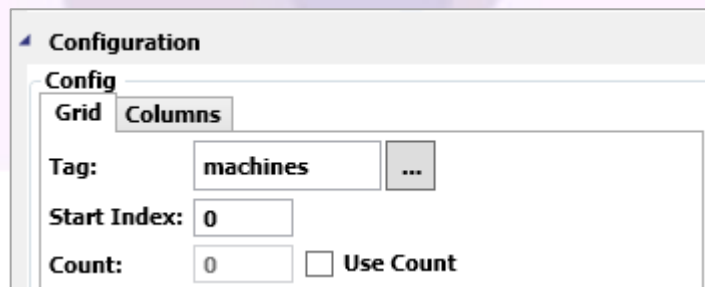
3.1.1. How to Add a Tag to the Object

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, as shown in the image below:



3.1.2. Start Index

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):



3.1.3. Tag Structure

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"

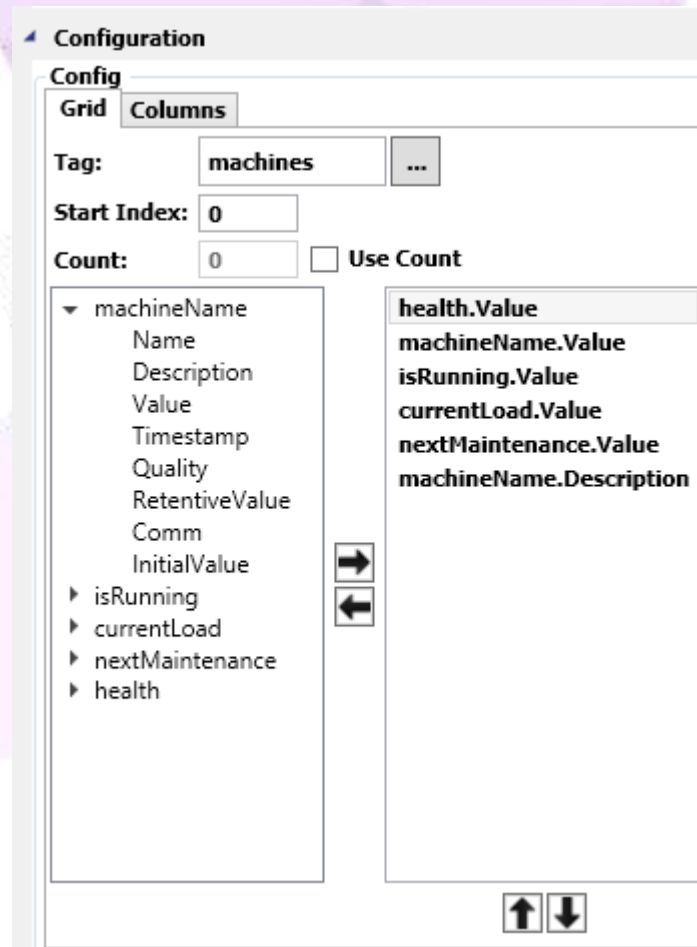
DataType tag was added with inner tags: health, machineName, etc.

3.1.4. Columns: Add, Remove, Change Order

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

3.1.5. Example

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 look at the column configuration options described in the previous sub-chapter.

3.3.1. Using a Dynamic Image

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).

Configuration

Config

Grid Columns

Path: health.Value

Appearance

Header: Health Type: Image

Width: 100 Input enable

Visible: true

Header Align: Center

Logic

Is Dynamic Image

Dynamic Image

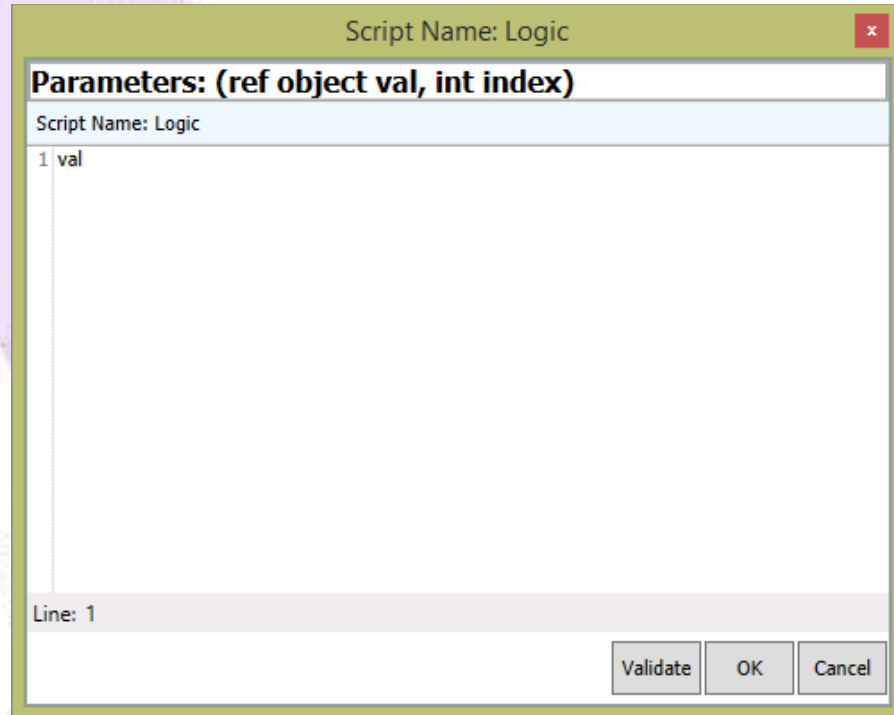
Default Image

Expression: val

Value	Image
0	
1	
2	

Source File

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 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.

3.3.2. Display Simple Text Where Input Disabled

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.

The screenshot shows the Configuration dialog box for a MultiTagViewer column. The dialog is titled "Configuration" and has a "Config" tab. The "Grid" tab is selected, and the "Columns" section is active. The "Path" is set to "machineName.Value". The "Appearance" section includes the following settings:

- Header: Name
- Type: Text
- Width: 170
- Input enable:
- Visible: true
- Select Text On Focus:
- Background Color: (empty)
- Foreground Color: (black)
- Header Align: Center

The "Logic" section includes the following settings:

- Text: val
- Mouse Down: (empty)
- Mouse Up: (empty)
- Mouse While: (empty)
- Mouse Double Click: (empty)

3.3.3. Displaying a Checkbox Represent Boolean

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

Configuration

Config

Grid **Columns**

Path: isRunning.Value

Appearance

Header: Is Running Type: CheckBox

Width: 140 Input enable

Visible: true

Background Color: ...

Foreground Color: ...

Header Align: Center

Logic

Text: val = (@machines[ir ...

Tag: val ...

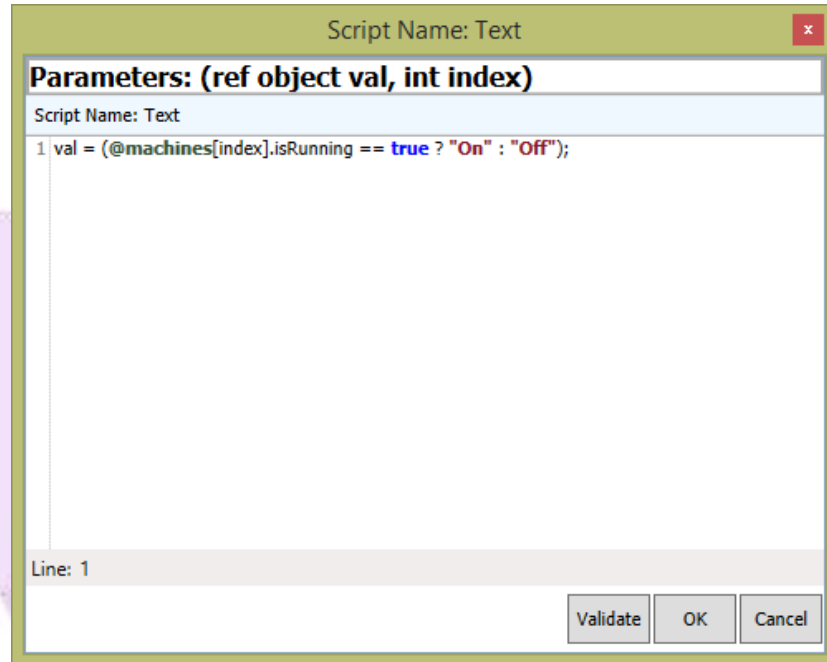
Mouse Down: ...

Mouse Up: ...

Mouse While: ...

Mouse Double Click: ...

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 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.

3.3.4. Concatenate or Link the Tag's Value

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.

Configuration

Config

Grid **Columns**

Path: nextMaintenance.Value

Appearance

Header: Next Maintenance Type: Text

Width: 150 Input enable

Visible: true

Select Text On Focus

Background Color: ...

Foreground Color: ...

Header Align: Left

Logic

Text: val = val + " (days)" ...

Mouse Down: ...

Mouse Up: ...

Mouse While: ...

Mouse Double Click: ...

3.3.5. Column Changed to a Button Type

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.

Configuration

Config

Grid Columns

Path:

Appearance

Header: Type:

Width: Input enable

Visible:

Background Color: ...

Foreground Color: ...

Header Align:

Logic

Text

Mouse Down

Mouse Up ...

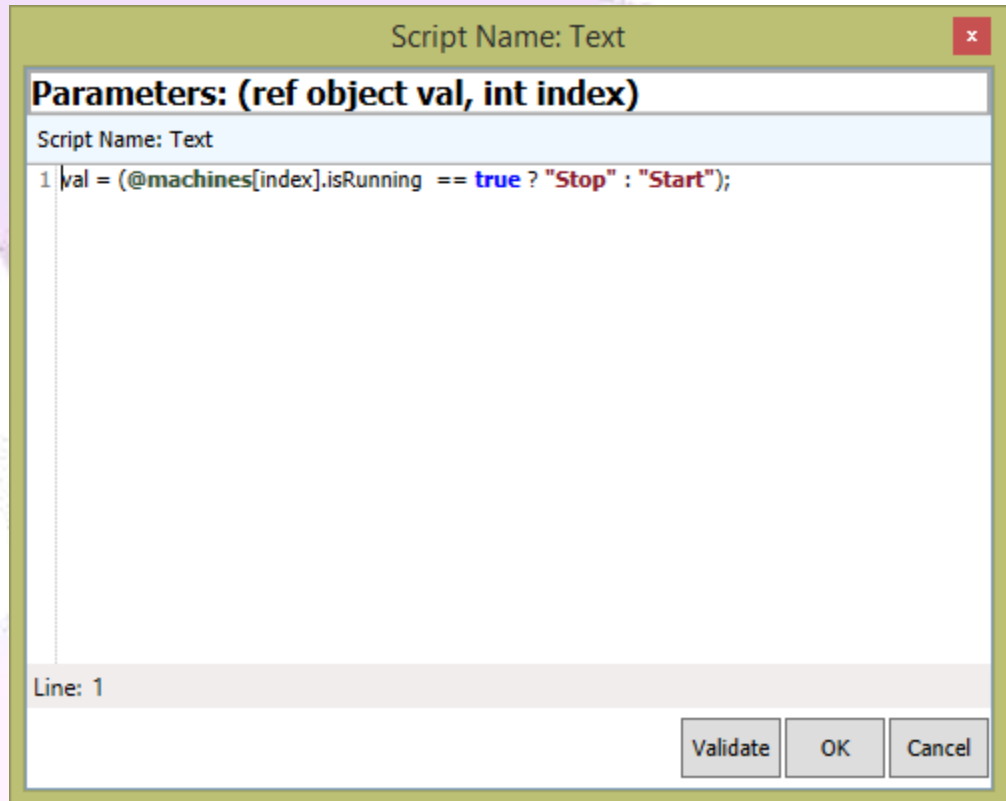
Mouse While ...

Mouse Double Click ...

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

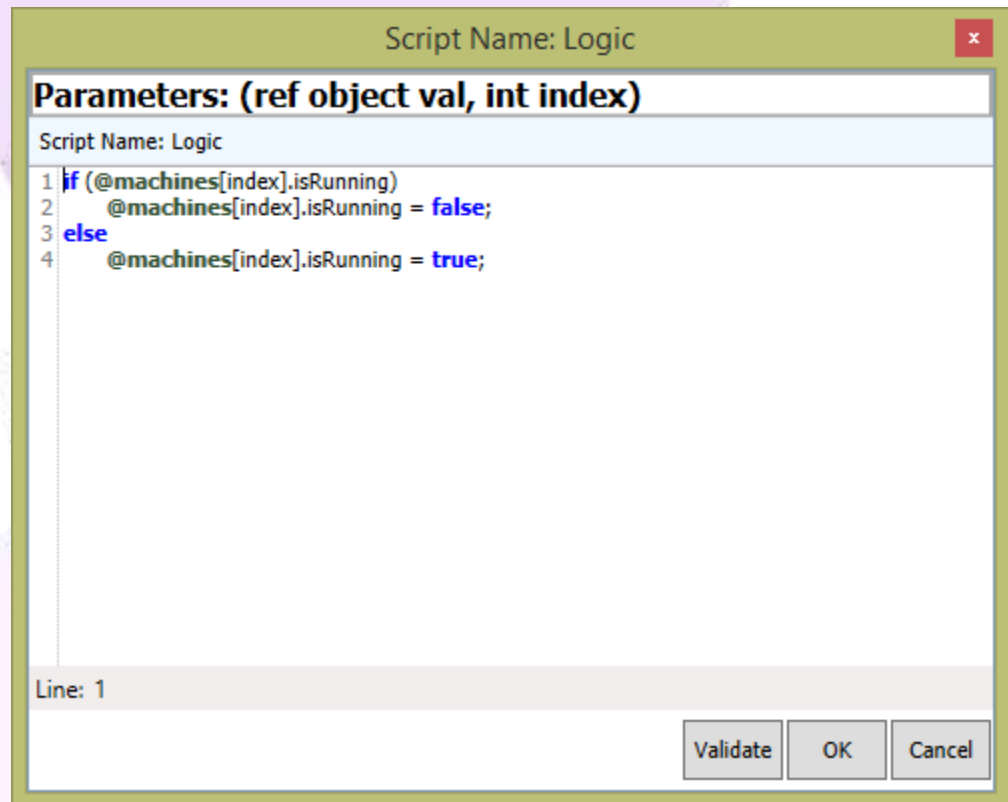
Button Text Expression:

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

3.4.1. Header

The column name that will be displayed in the MTV.

3.4.2. Type

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

3.4.3. Width

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

3.4.4. Input Enable

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

3.4.5. 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 on the result.

3.4.6. Select Text on Focus

Clicking a cell will select the text.

3.4.7. Is Dynamic Background

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

3.4.8. Background Color

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

TagInt.Name	TagInt.Value
Tags/TagDT[0].TagInt	0
Tags/TagDT[1].TagInt	1

Background Color: Aqua

3.4.9. Foreground Color

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

TagInt.Name	TagInt.Value
Tags/TagDT[0].TagInt	0
Tags/TagDT[1].TagInt	1

Foreground Color: Red

3.4.10. Header Align

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

Left

Center

Right

TagInt.Name
Tags/TagDT[0].TagInt
Tags/TagDT[1].TagInt

TagInt.Name
Tags/TagDT[0].TagInt
Tags/TagDT[1].TagInt

TagInt.Name
Tags/TagDT[0].TagInt
Tags/TagDT[1].TagInt

3.4.11. Logic

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

3.4.12. Text

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

3.4.13. 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	Segoe UI <input type="text" value="12"/>
	<input type="checkbox"/> B <input type="checkbox"/> <i>I</i>
Selected Index	<input type="text" value="3"/>
Row Height	<input type="text" value="25"/>
Cells Spacing	<input type="text" value="10"/>
Show Header	<input checked="" type="checkbox"/> True
Show Grid Lines	<input type="checkbox"/> False
Number of decimals	<input checked="" type="checkbox"/> Use Default <input type="text" value="2"/>

3.4.14. Font

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

3.4.15. 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.

3.4.16. Row Height

Determine or change the object's row height.

3.4.17. Cells Spacing

Determines the distance between cells inside the MultiTag.

3.4.18. Show Header

Determine to hide/show the object's header.

TagInt.Name	TagInt.Value
Tags/TagDT[0].TagInt	0
Tags/TagDT[1].TagInt	1

Tags/TagDT[0].TagInt	0
Tags/TagDT[1].TagInt	1

Show Header: True

Show Header: False

3.4.19. Show Grid Lines

Determine to hide/show the object's grid lines.

TagInt.Name	TagInt.Value
Tags/TagDT[0].TagInt	0
Tags/TagDT[1].TagInt	1

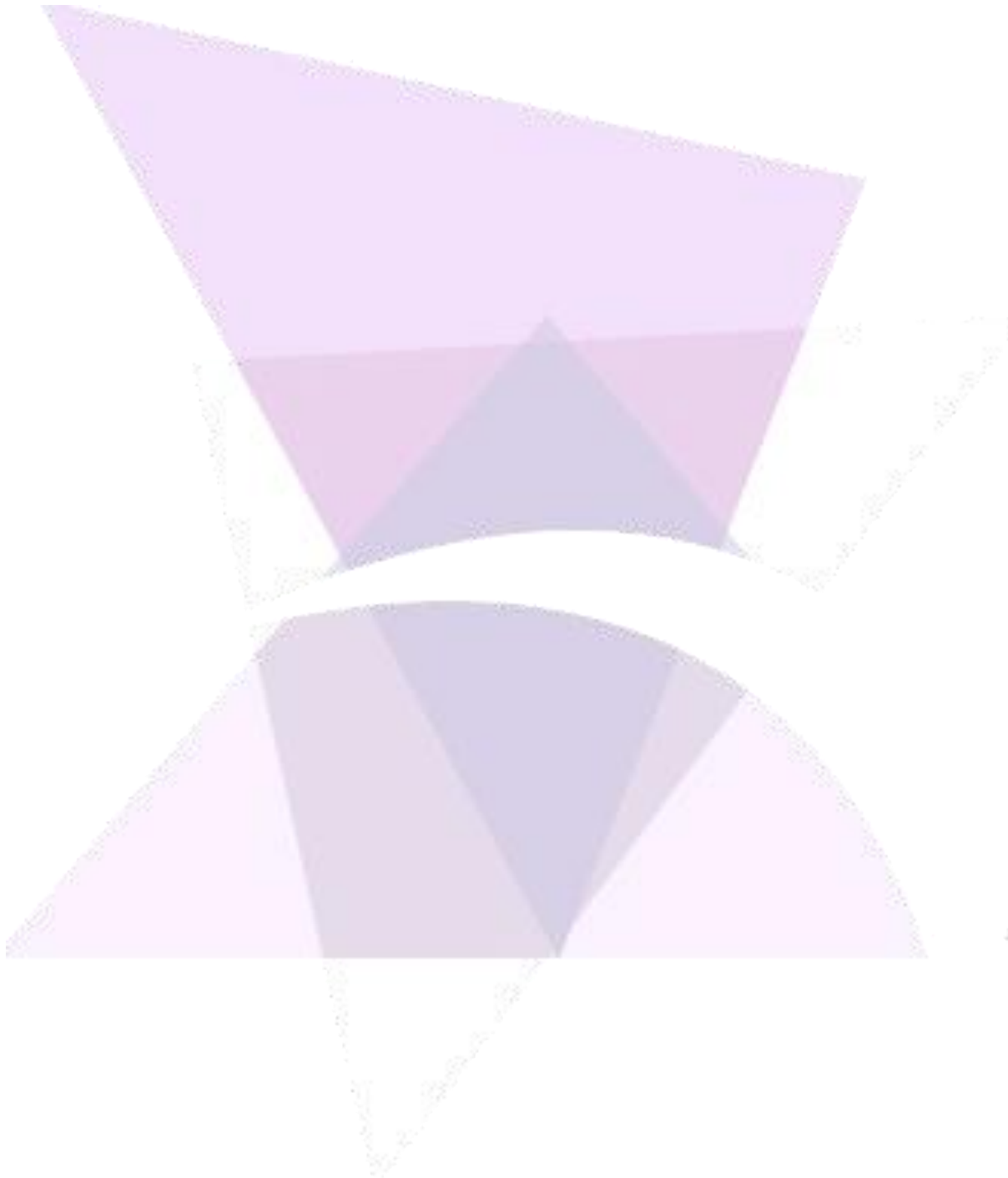
TagInt.Name	TagInt.Value
Tags/TagDT[0].TagInt	0
Tags/TagDT[1].TagInt	1

Show Grid Lines: False

Show Grid Lines: True

3.4.20. Number of Decimals

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



ADISRA®, InsightView™, and KnowledgeView™ are the registered trademarks of ADISRA, LLC.

© 2022 ADISRA, LLC. All Rights Reserved.

ADISRA · 3432 Greystone Drive, Suite 125 · Austin, TX 78731
Phone: 1-833-5ADISRA (1-833-523-4772)

www.ADISRA.com