



Medialon Marquee

User Guide

Trademark Information

Medialon is a trademark of 7thSense Design Ltd. Product or company names that mentioned in 7thSense Design Ltd publications are tradenames or trademarks of their respective owners, and such trademarks may also be registered in their respective countries.

Copyright Information

All Rights Reserved. This document is copyrighted © by 7thSense Design Ltd and shall not be reproduced or copied without express written authorization from 7thSense Design Ltd.

The information in this document is subject to change without notice. 7thSense Design Ltd assumes no responsibility for errors, and/or omissions contained in this information.

Document Revision

Date	Document edition	Software version	Revision details	Author/Editor
October 2020	1	0.1	New release	Liam Neale
May 2021	2	0.1	Revised logo, illustration	Andie Davidson

M616-2

Contents

1 Introduction	3
1.1 Overview of Marquee	3
2 Marquee Environment	4
2.1 The Three Building Views	4
2.2 Controls	5
2.3 The Edit View	6
2.4 The Inspector	6
2.5 Menu	8
2.5.1 File	8
2.5.2 Edit	9
2.5.3 View	11
2.5.4 About	11
2.6 Status Panel	12
2.6.1 Project Information	12
2.6.2 Manager Connection Status	12
2.7 Page View	12
2.7.1 Adding Widgets	13
2.7.2 Removing Widgets	14
2.7.3 Editing Widgets	14
2.8 Preview and Run Modes	14
3 Widgets	16
3.1 Button	17
3.2 Lists	19
3.3 Text/TextField	20
3.4 Progress Bar	21
3.5 Sliders	22
3.6 Toggle (On/Off)	23
3.6.1 Switch	23
3.6.2 Toggle	24
3.7 Custom	24
3.7.1 NDI	24
3.7.2 Keypad	24
3.8 LED	26
3.9 Groups	26

3.9.1 Standard Group.....	26
3.9.2 Grid Group.....	27
3.10 Notifications	29
3.11 Gallery.....	30
4 Connecting to Manager.....	31
4.1 Set Up.....	31
4.1.1 Connection.....	31
4.1.2 Set up Shared Groups.....	32
4.2 Connect Widgets to Shared Variables/ Tasks	33
4.2.1 Manager Variables.....	33
4.2.2 Manager Tasks.....	35
4.3 Marquee Syntax for Single Instruction.....	35
4.3.1 Internal Attribute Communication	35

Medialon EULA and Warranty

Please refer to document M561 [Medialon EULA and Warranty](#) on our website.

1 Introduction

Marquee is an optional extension to Medialon Manager and Showmaster for generating a UI/Widget interface (as an alternate to Manager/Showmaster Local Panels and WebPanels) in external standalone application.

1.1 Overview of Marquee

Marquee has two way communication with Manager/Showmaster where users have the ability to manipulate shared variables, shared tasks and shared devices from Marquee.

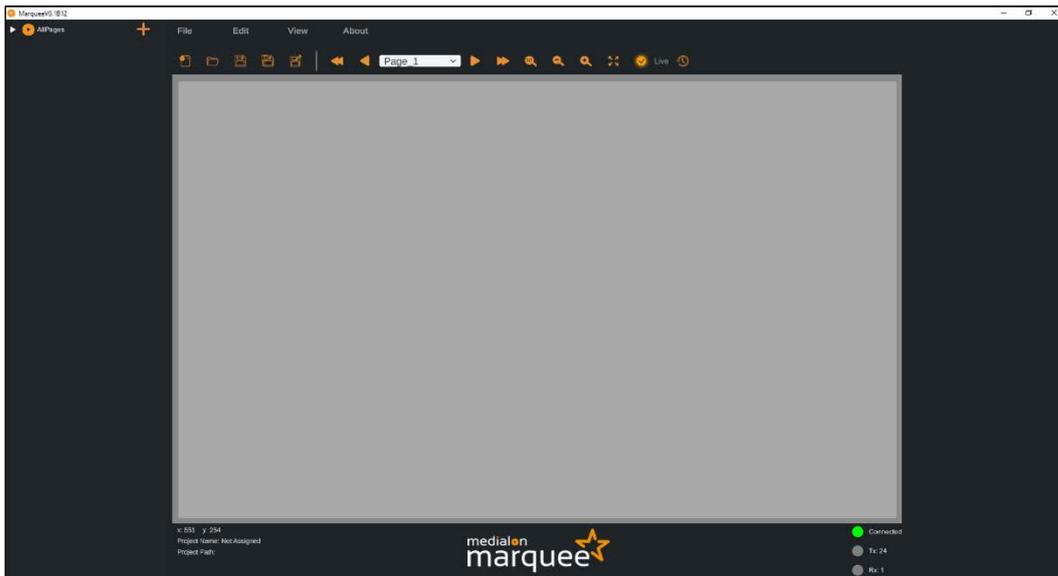
Marquee includes all the Graphical User features that Manager has, as well as:

- More types of widgets
- The ability to duplicate widgets and keep their attributes
- Live editing of widgets while the project is communicating with Manager/Showmaster
- Ability to add user levels
- Edit multiple widgets simultaneously
- Undo-Redo stack

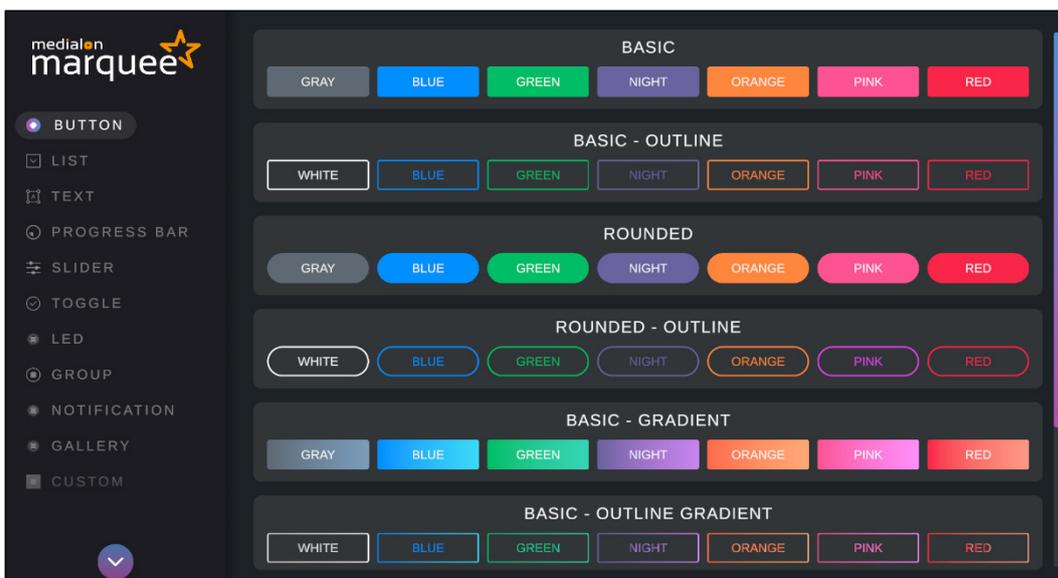
2 Marquee Environment

2.1 The Three Building Views

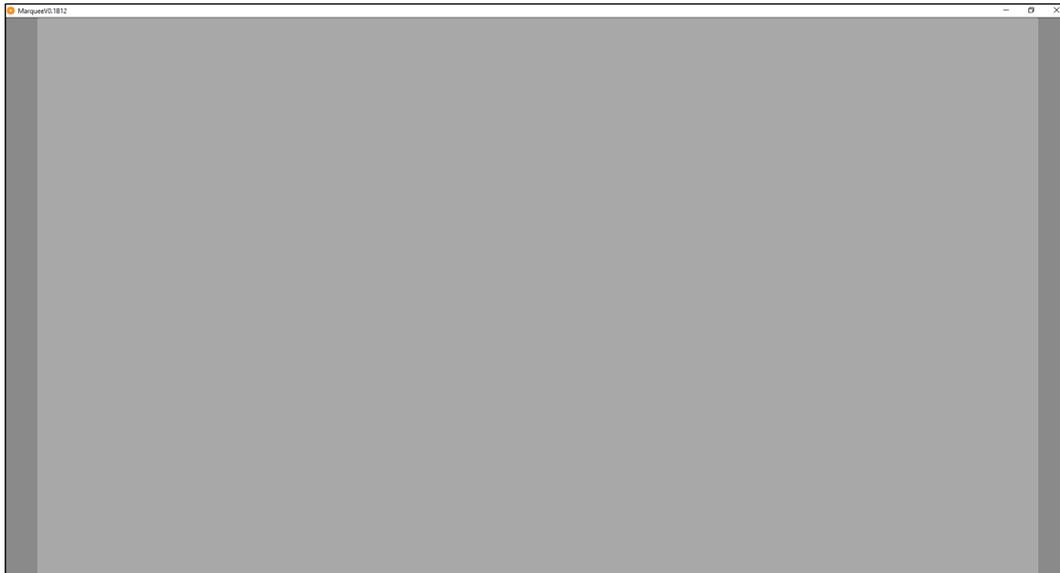
When you first open Marquee, the view that you are in is the 'Edit View'. This is where you will spend most of your time. Press F11 to toggle fullscreen mode.



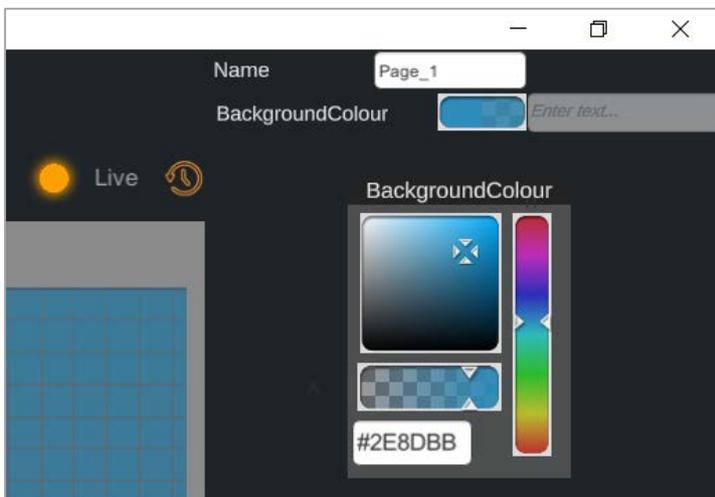
If you then press [Esc] it will take you to the 'Picker View'. This is where you can add in pre-built widgets as well as play with them, to see which ones you want to use.



Tapping the [Esc] key again takes you back to Edit View (nothing has yet been added to the canvas). Tapping [F5] switches the interface to 'Run View'. This is what the end user would see.



The background colour of this view is configurable when the Page is selected in Tree View:



2.2 Controls

Edit View

- [Esc] to get to Picker View.
- Left-Click & Hold to drag widgets. If multiple widgets are selected, hold Left[Ctrl] while dragging to affect all selected widgets.
- Hold [Alt] when dragging and sizing to release grid-snap.
- Left[Ctrl]+[D] duplicates the selected widgets.
- Hold Scroll Wheel & move mouse to pan around the canvas.
- Scroll Wheel zooms in and out of the canvas.
- Right-Click & Hold to draw a 'Rubber Band' selection box.

- [F5] to go to Run View.
- [Ctrl]+[Z] to undo.
- [Ctrl]+[Y] to redo.

Picker View

- [Esc] to go to Edit View.
- Left[Ctrl] & Left-Click to insert the selected widget on the canvas.
- Left-Click on left tabs to browse different types of widget.

Run View

- [F5] to go back to Edit View.

2.3 The Edit View

The Edit View is built up of 5 panels: Tree View, Menu, Page Area, Inspector, and Status Panel.

Tree View

The Tree View is positioned on the left side of the screen and is where any widgets in your scene will be stored for you to access. The top level (AllPages) cannot be removed or re-parented. The child of AllPages must be a Page type widget. Any type of widget can be a child of a page, including another page.



In this scene there are two pages. Page_1 has three widgets and Page_2 has zero.

Button_0 has an LED as a child.

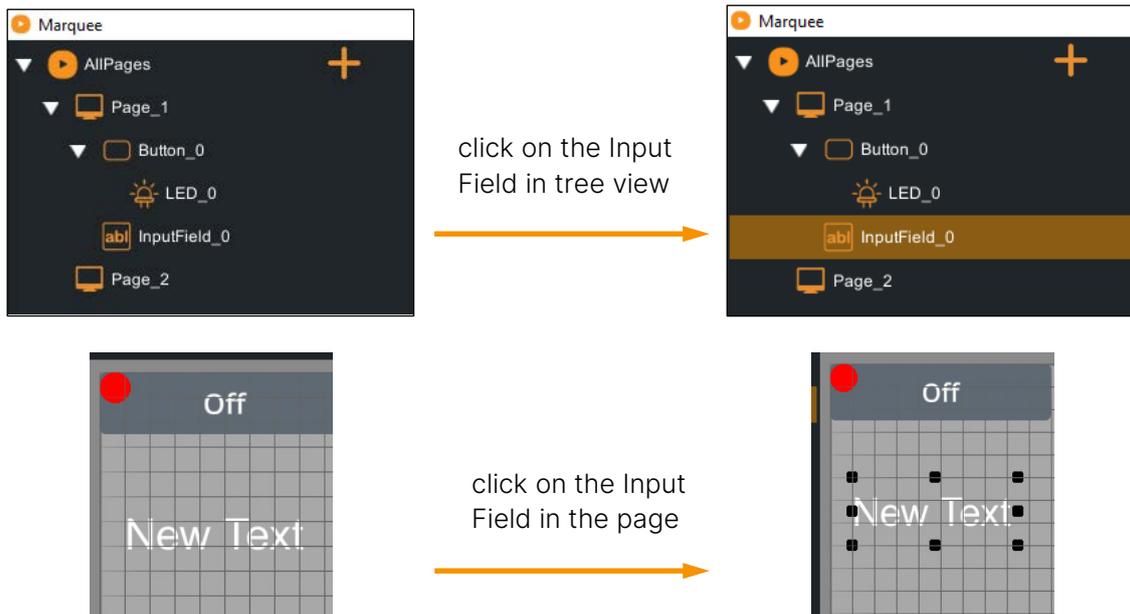
You can use your mouse to drag the nodes in the Tree View to different parts of the tree except for the AllPages root node. You also cannot place a page in a non-page widget.

All nodes also have an icon representing the widget's type.

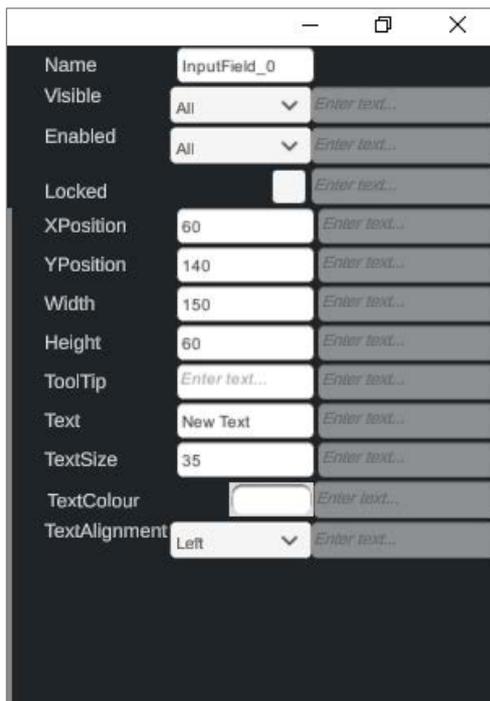
2.4 The Inspector

To open the Inspector, either click on any widget in the tree view or on the widget in the centre canvas editing area. The Inspector is where you can edit widget attributes, and supports editing parameters of multiple widgets if they are multi-selected in the Tree View or with a Rubber Band selection.

To display the Inspector



Either method brings up Inspector for an input field widget on the right hand side of the screen:



There are four types of Inspector fields:

1. Edit boxes into which you can type.
2. Tick boxes for On/Off.
3. Drop-downs for fixed selections.
4. Colour picker to select colour. The colour picker is only displayed when you click on a colour type attribute.

Multi-Select:

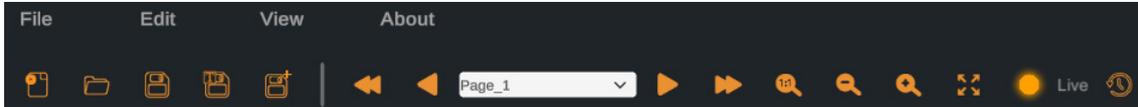
- When multiple widgets are selected, the only attributes displayed in the Inspector are those shared by all the selected widgets.
- If attribute values are different, an ellipsis (...) is displayed, or for colour attributes, the colour is greyed-out.

Shared Vars:

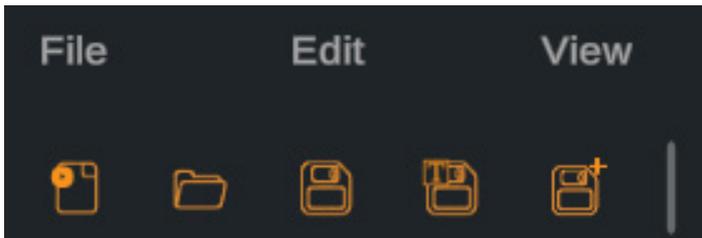
On the right of each attribute, a shared variable from the connected Manager/Showmaster project can be connected to the attribute.

2.5 Menu

At the top of Edit View is the main menu and two toolbars, for file operations and page operations:

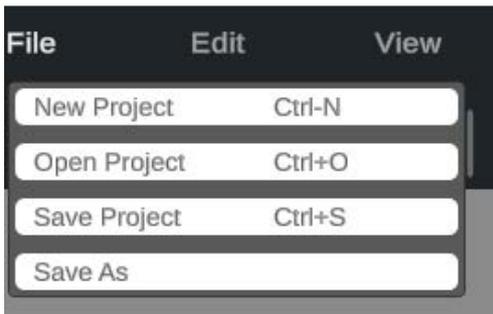


2.5.1 File



New Open Save Save as Save plus

The File drop-down menu is similar to the toolbar and shows the shortcuts:



New Project: creates a new empty project with no connection attributes.

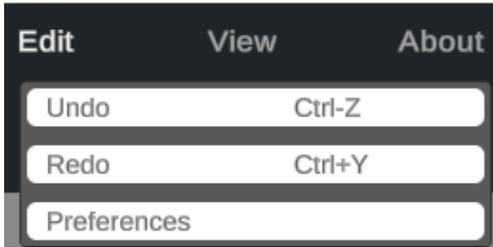
Open Project: opens a previously saved Marquee project.

Save Project: saves (overwrites) the loaded project with the current configuration.

Save As: opens a dialog box for saving the project to a new file/location.

Save Plus: saves a new time-stamped file copy in the project's \Backups folder.

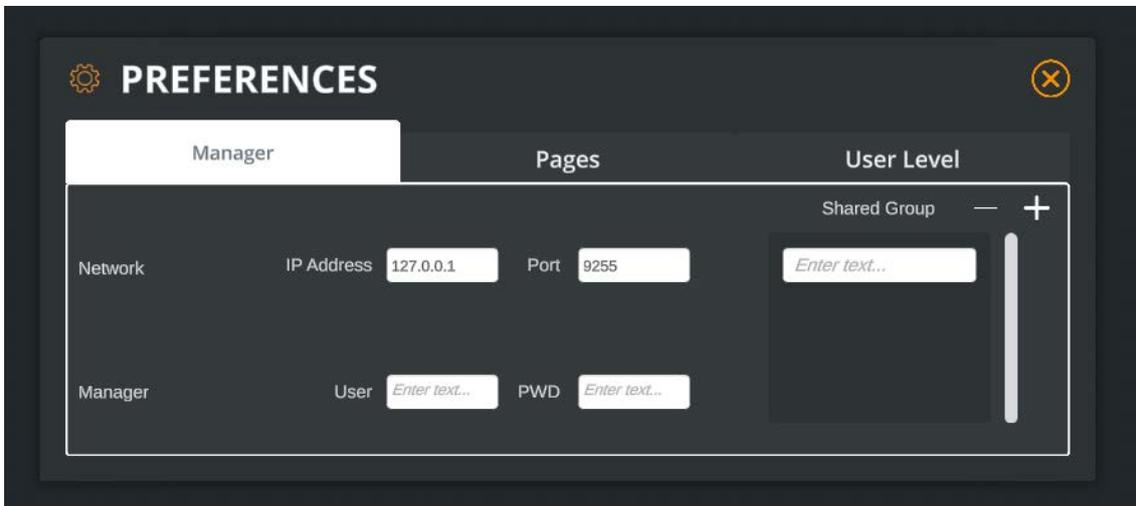
2.5.2 Edit



Preferences

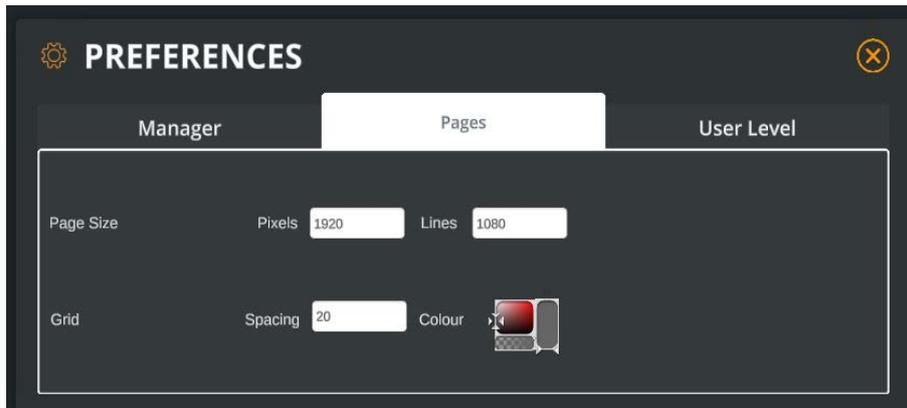
In *Edit > Preferences* there are 3 tabbed dialogs: Manager, Pages, User Level.

Manager



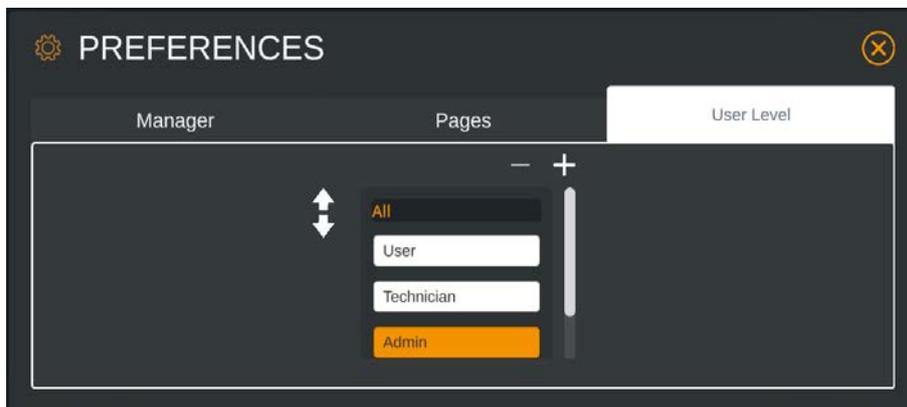
This is where you point Marquee to a Medialon Manager or Showmaster instance to which it should connect, by entering the IP address and Port (default 9255). Then enter the names of the Shared Groups you want Marquee to be allowed to access in the Shared Group list, by clicking the plus (+) button to add a new Shared Group. To remove a Shared Group from the Marquee project, select its name and click the minus (-) button.

Pages



Set the page resolution, grid spacing (1 or 0 will remove the grid), and grid colour.

User Level



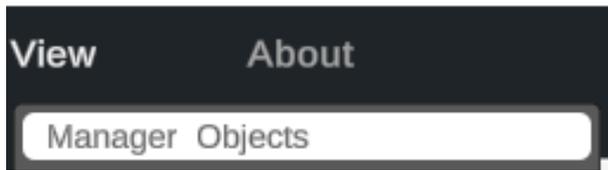
Here you can create global user access levels for the Marquee project, which can be assigned to individual widgets. The top of the list (All) has the least amount of user access. Widgets which have not specifically been assigned a user access level default to 'All'.

Users who log in with increasing levels of access are able to view and interact with the widgets tagged at or below the logged-in user's level. (See [PIN Keypad](#) for more information about user access levels.) In the example above, a Technician can see everything a User can, plus everything a Technician is allowed to see, but cannot see the widgets tagged for Admin.

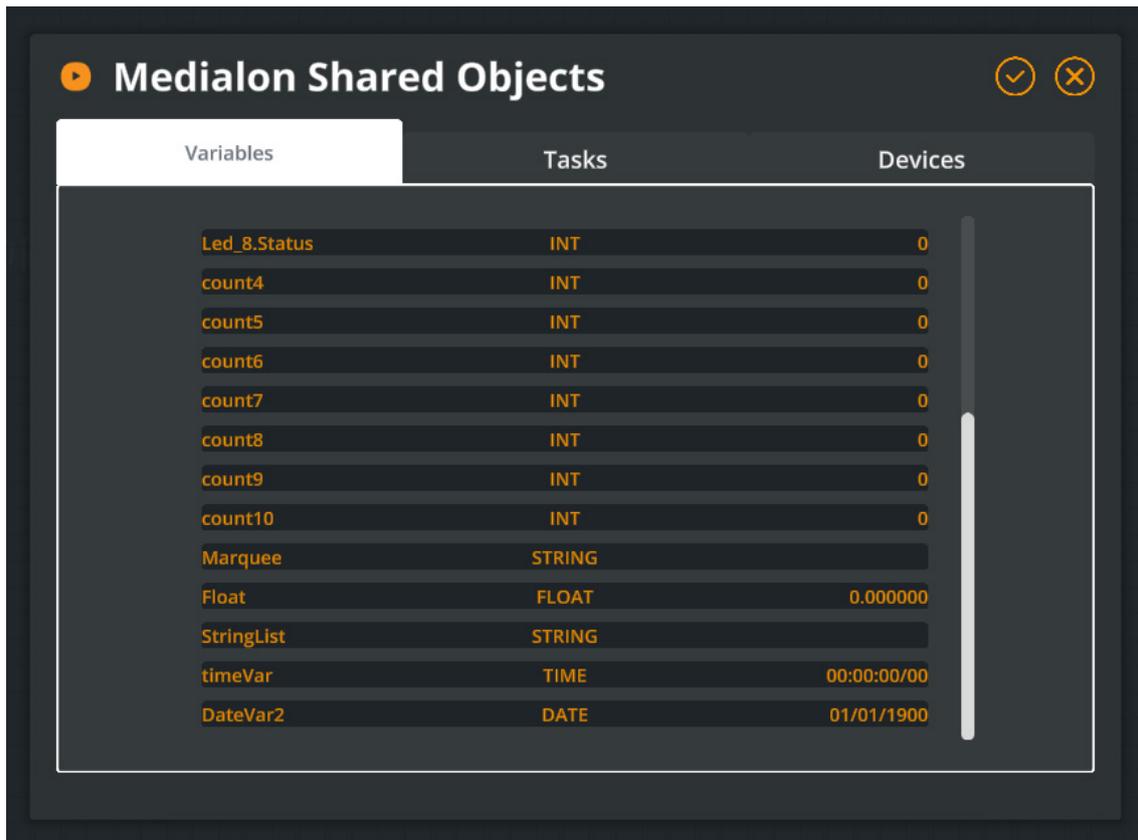
To add a user access level, click the plus (+) button and give the user access level a name. To remove the selected user access level, click the minus (-) button.

To move a user access level up or down in priority, select the name and click the up or down arrows to rearrange the list.

2.5.3 View



Select *View* > *Manager Objects* to view all of Manager's (or Showmaster's) Shared Variables, Tasks and Devices:



The screenshot shows a window titled 'Medialon Shared Objects' with a play button icon and a close button. It contains three tabs: 'Variables', 'Tasks', and 'Devices'. The 'Variables' tab is active, displaying a table with the following data:

Variables	Tasks	Devices
Led_8.Status	INT	0
count4	INT	0
count5	INT	0
count6	INT	0
count7	INT	0
count8	INT	0
count9	INT	0
count10	INT	0
Marquee	STRING	
Float	FLOAT	0.000000
StringList	STRING	
timeVar	TIME	00:00:00/00
DateVar2	DATE	01/01/1900

'Variables' include the names, types, and current values of all shared variables. 'Tasks' shows the names and types of shared Tasks.

2.5.4 About

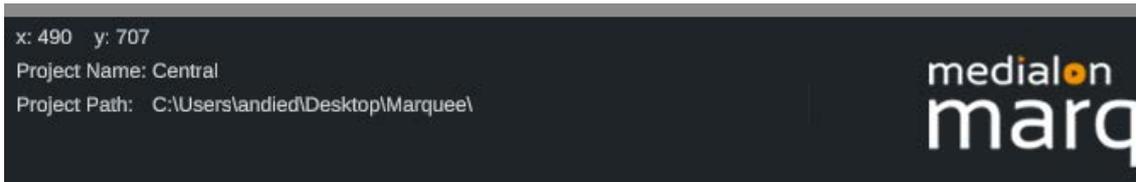
This menu item displays information about the version of the Marquee application and where you can go to find help or more information about Medialon.

2.6 Status Panel

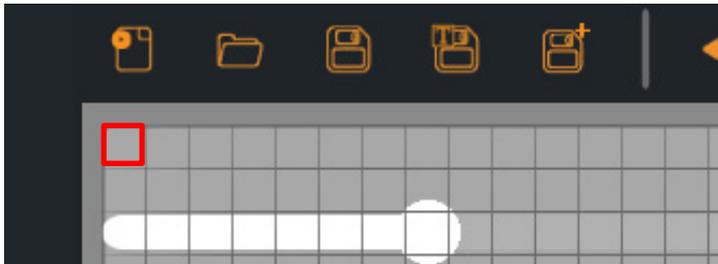
The Status Panel is located at the bottom of the screen and displays two types of information for the user.

2.6.1 Project Information

The first displays the name and path of the project currently being edited. If you are in an empty project then it will display 'Not Assigned'.



It also displays the X/Y coordinates of your mouse where (0,0) is top left of the Page View:



2.6.2 Manager Connection Status

At the bottom right of the screen there are three LEDs:



Connected: the connection status between Marquee and Manager. Green is successful, orange is connected at socket level but not OpenCAP, and red is failed.

Tx: flashes green to show Marquee has sent a message to Manager.

Rx: flashes green to show Marquee has received a message from Manager.

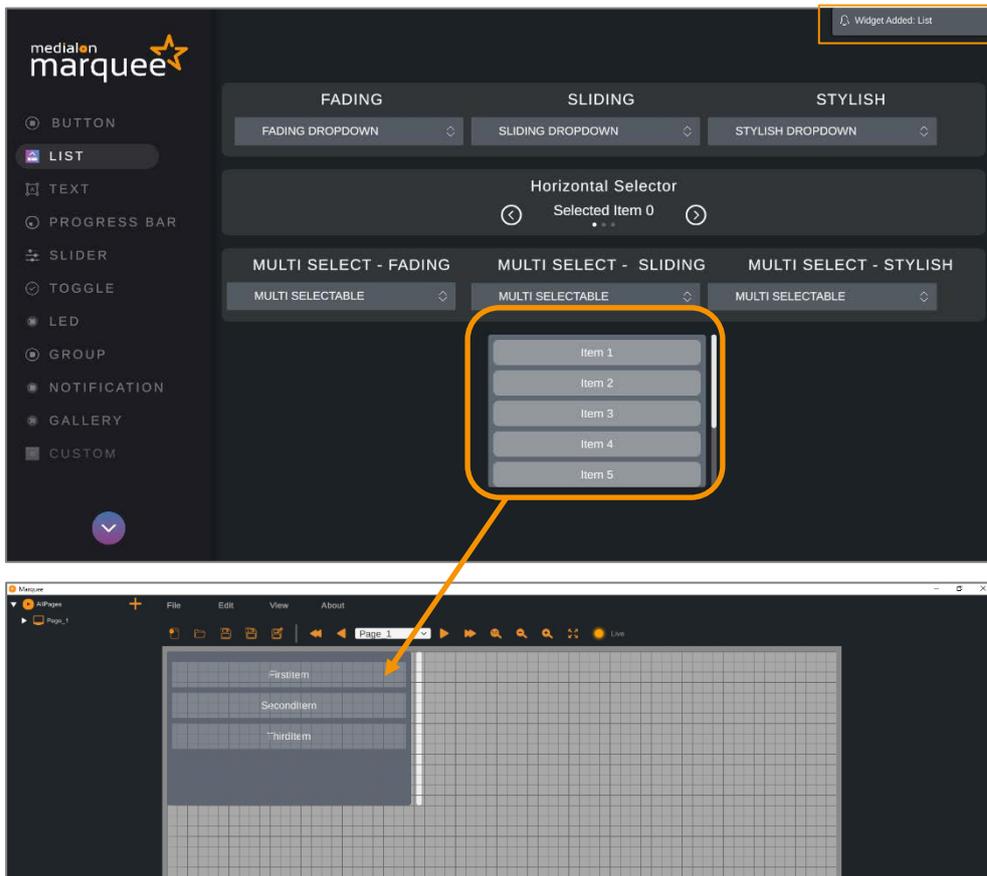
Note: if Live mode is not enabled, no Tx or Rx will happen.

2.7 Page View

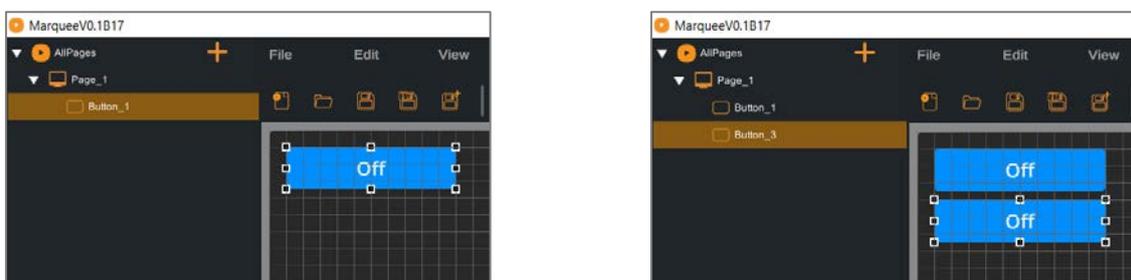
This is where the contents of the selected page are shown. If you are not in Live mode, the widgets are editable by selecting them with the mouse or drawing a 'rubber band' selection box using the right mouse button.

2.7.1 Adding Widgets

To add widgets to the page view, press [Esc] to go to the Widget Picker View and find a widget you want. To add the widget to the page, hold the left [Ctrl] button and click the mouse on the widget you wish to add. There will be a 'Widget added' confirmation in the top right that fades away. Press [Esc] again to see the widget on the page. Example, add a List widget:



Alternatively, you can select and use [Ctrl]+[D] to duplicate single or multiple widgets at once (along with their attributes). This means you can customize a widget style and duplicate it instead of having to modify the attributes for each widget. Note that duplicated widgets are placed at the same X/Y location as the original widget.



Select → [Ctrl]+[D] → attributes are retained and a Button is added in the tree

2.7.2 Removing Widgets

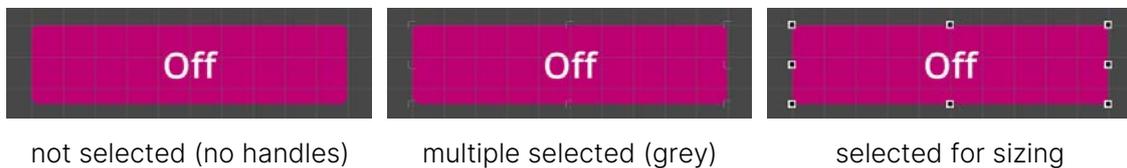
To remove a single or multiple-selected widgets, press [Ctrl]+[Delete] on the keyboard.

2.7.3 Editing Widgets

'Live' mode disables mouse-based position/size editing, so turn it off for the page before selecting, moving, and resizing widgets.

Hold the left mouse button to drag single selected items. [Ctrl] + click/drag to select or move multiple items. Widgets snap to the grid (*Edit > Preferences > Pages* for spacing); hold [Alt] to release from the grid.

For sizing, hover the mouse over a black handle, drag with the left mouse button. This applies to single widgets only. Mouse-based resizing is disabled when multiple widgets are selected.



Exact size and position can also be edited in the Inspector.

Attributes may be edited in Live mode by selecting the widget in the Tree View and modifying the attributes in the Inspector.

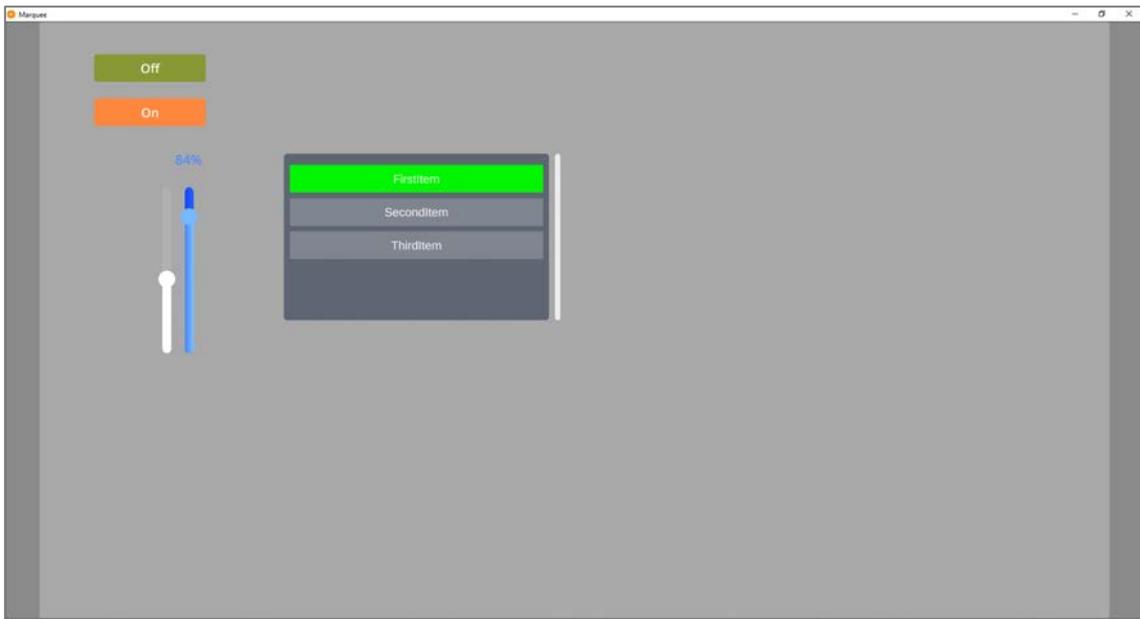
2.8 Preview and Run Modes

To preview and test your Marquee project, use the toolbar 'Live' toggle. This activates all the widgets in your Page View and disables mouse-based editing features, but keeps Marquee in an editing mode so you can navigate the Tree View and edit attributes in the Inspector.



To go into a 'full screen' version of your Marquee project, press [F5] to go into Run Mode. This mimics the full screen view of the project that the end user will see when the project is run in Marquee Player.

Marquee Environment



3 Widgets

Widgets are the UI elements that can be put into the edit view and given required attributes. Some attributes are shared by all widgets, some are specific to the type. Common attributes are:

Name	Button_1	
EnabledToggle	<input checked="" type="checkbox"/>	Enter text...
Visible	All	Enter text...
Enabled	All	Enter text...
Locked	<input type="checkbox"/>	Enter text...
XPosition	400	Enter text...
YPosition	200	Enter text...
Width	200	Enter text...
Height	55	Enter text...
ToolTip		Enter text...

Name: the name of the widget in the scene, default is the type of widget with an underscore and the next number as added in sequence (e.g. Button_3).

Visible: (String, Int) Permission to see the widget. If user levels have been setup, they will be available in the drop-down and can be selected for the widget. Only users with the selected level of access (or higher) will see the widget.

Enabled: (String, Int) Permission to interact with the widget. If user levels have been setup, they will be available in the drop-down and can be selected for the widget. Only users with the selected level of access (or higher) will be able to interact with the widget.

Locked: (String, Int) Tickbox, when enabled, the widget is locked from being selected, moved, or resized by the mouse. Attributes can still be adjusted by selecting the widget in the Tree View and modifying the attributes in the Inspector.

XPosition: (String, Int, Float) X-axis position of the widget, relative to the top left of the page.

YPosition: (String, Int, Float) Y-axis position of the widget, relative to the top left of the page.

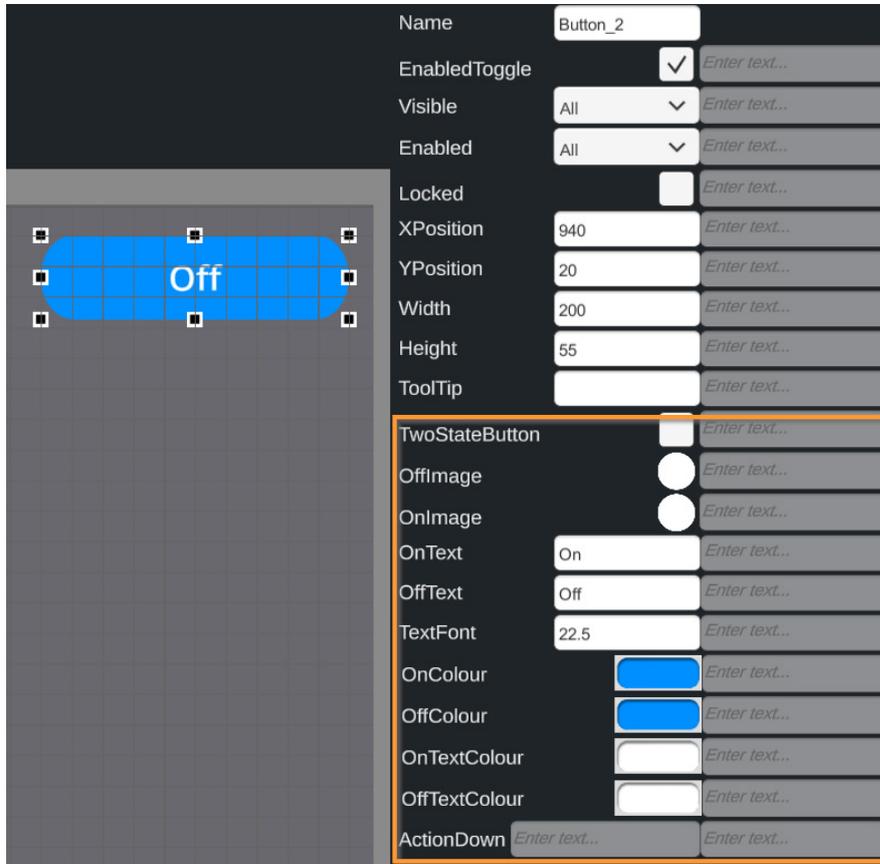
Width: (String, Int, Float) Width (in pixels) of the widget

Height: (String, Int, Float) Height (in pixels) of the widget

ToolTip: text revealed when hovering the mouse over the widget

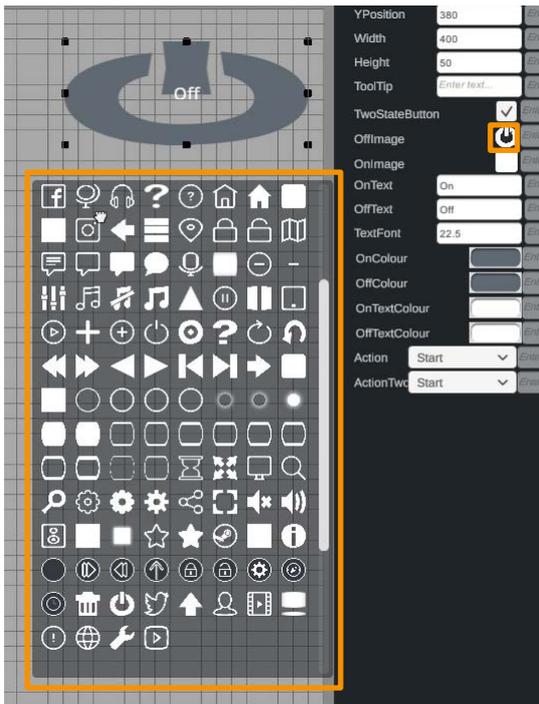
3.1 Button

Below Tooltip on the Image below, you can see the additional attributes for a **button**.



TwoStateButton: (String, Int) Tickbox that changes the button from one-state (normal) to two-state (toggle). If the button is two-state then a second 'ActionUp' attribute is added to the bottom of the list. *How it works:* If the button is triggered from off to on, then the 'ActionDown' attribute is triggered. If the button is switched from on to off, the 'ActionUp' attribute is triggered.

OnImage: (String) Image to fill the button when in its Off state. Apply as for OffImage.



OffImage: (String) Image to fill the button when in its Off state. Click the current image to the right of the text for a selector panel of available image icons. You may need to adjust the button dimensions to achieve the correct aspect ratio for the image.

OnText: (String, Int, Float) Text to display in the button when in the on state.

OffText: (String, Int, Float) Text to display in the button when in the off state.

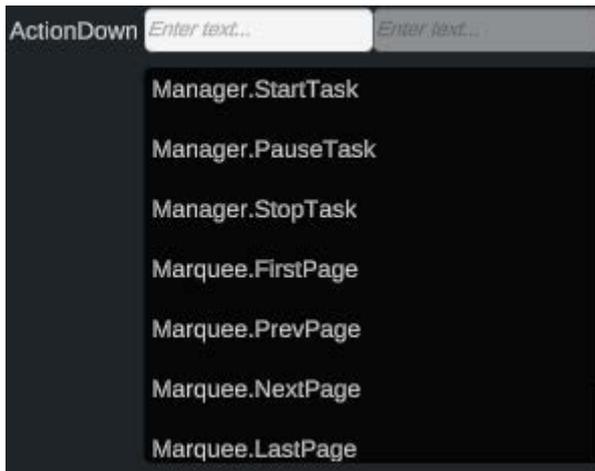
OnColour: (String) Changes the colour for the on state. Depending on the type of button, it changes how the colour is affected; the gradient buttons have two on and off colours.

OffColour: (String) Changes the colour for the off state.

OffTextColour: (String) Changes the colour of the off state text

OnTextColour: (String) Changes the colour of the on state text

Action: (String) When the button is pressed this action is triggered. Some actions are internal, others include: start, stop, pause task in Manager. The dropdown offers these actions:



StartTask: Starts the corresponding task name in the input field to the right in Manager.

PauseTask: Pauses the task in *Manager*

StopTask: Stops the task in *Manager*

FirstPage: goes to first page in *Marquee*

PrevPage: goes to previous page in *Marquee*

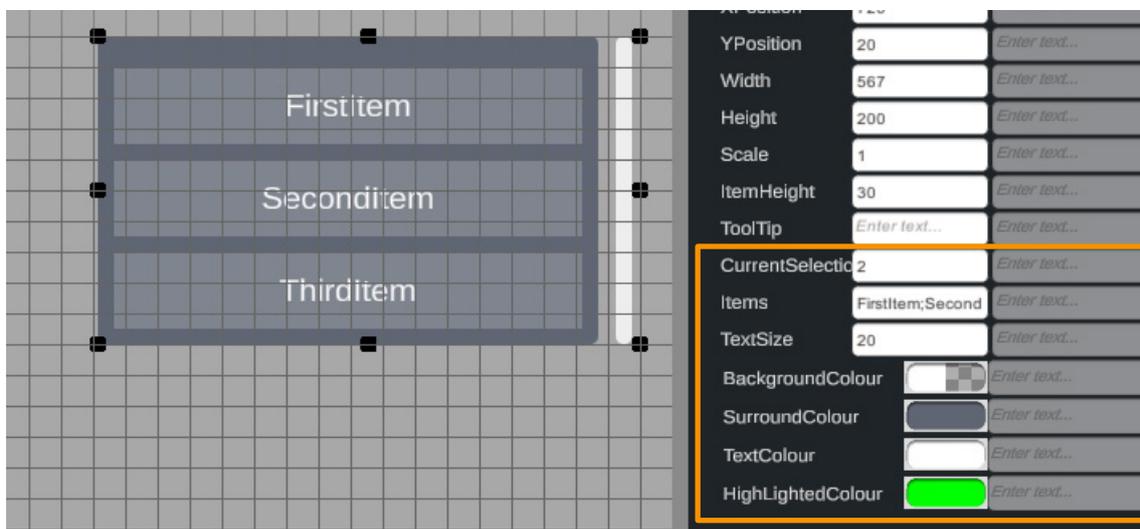
NextPage: goes to next page in *Marquee*

LastPage: goes to last page in *Marquee*

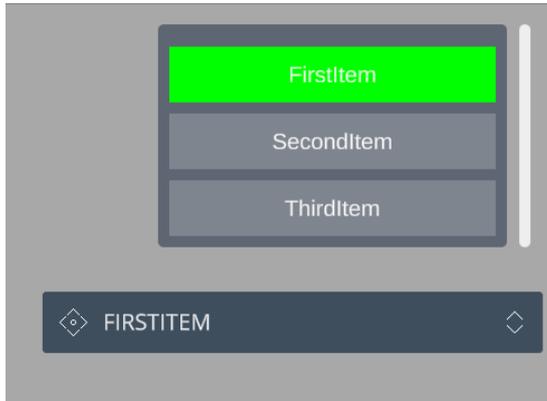
GoToPage: goes to a page by name in *Marquee*.

In the example above, TwoStateButton is selected so ActionDown has been added as an attribute.

3.2 Lists



CurrentSelection: (String, Int, Float, Enum) Shows which item in the list is currently highlighted or showing, depending on the type of list:



Items: (String, Enum, semicolon-delimited list of each item in the list) If current selection is Enum type, this attribute is not available.

TextSize: (String, Int, Float) Size of the text displayed in list items.

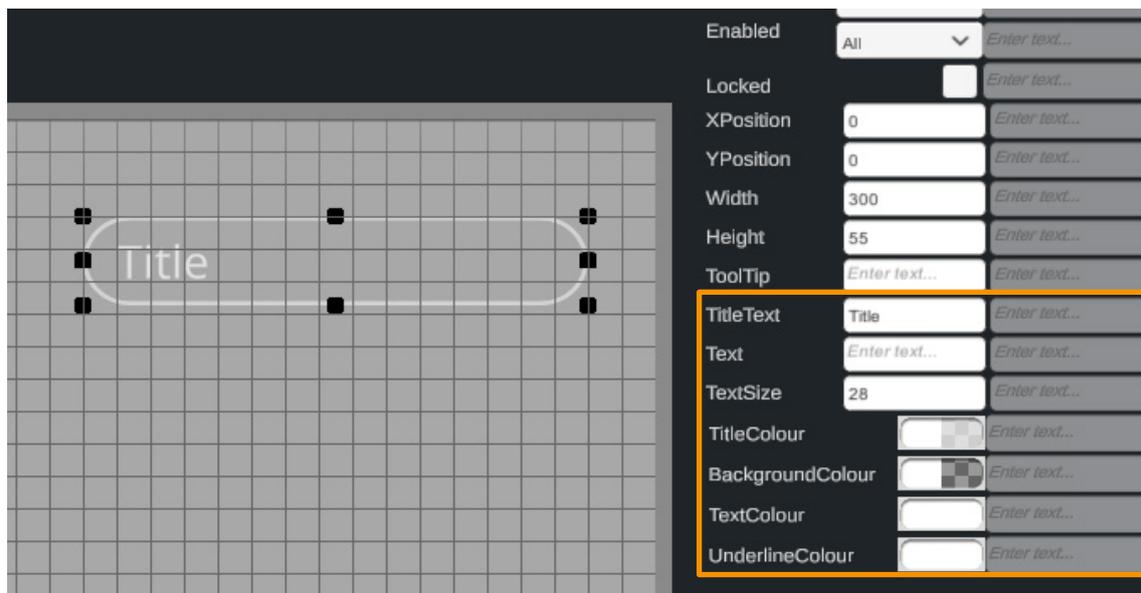
BackgroundColour: (String) Colour behind unselected list items.

SurroundColour: (String) Colour of the list background.

TextColour: (String) Colour of the list item text.

HighlightedColour: (String) Colour of the selected list item.

3.3 Text/TextField



This widget type can be used to display text or to provide a text input field for users to type in string values.

Widgets

Text: (Int, String, Float) Text string value to appear in the input field.

TextSize: (String, Int, Float) Size of the text in the text string.

TextColour: (String) Input field text colour.

TitleText: (String, Int, Float) Text for the *title* of the input field; *not all text widgets have this attribute*. Some widgets display the Title when the Text attribute is null, other widgets display the Title above the Text.

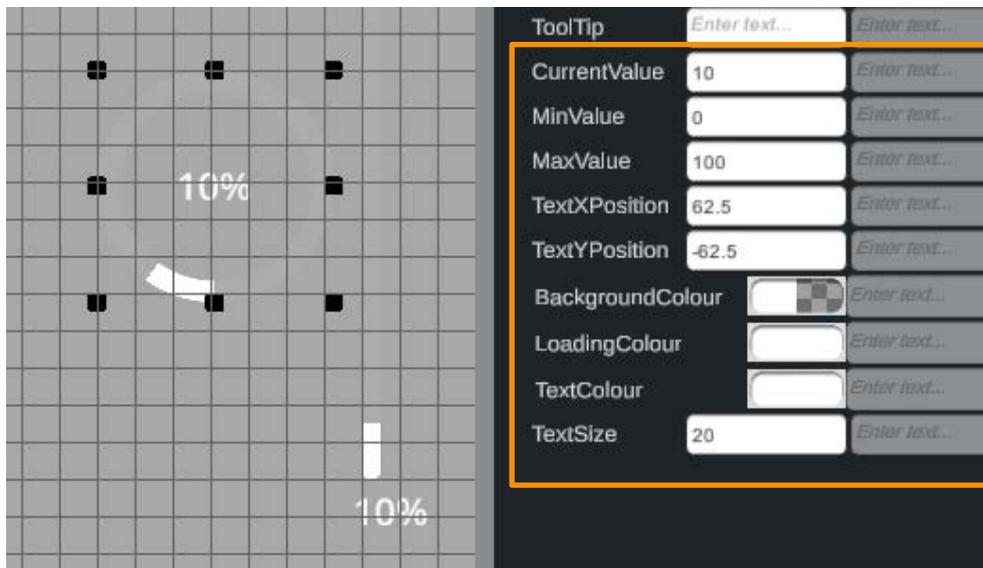


TitleColour: (String) Colour of the title text.

UnderlineColour: (String) Colour for the underline or outline of the input field (see illustration).

TextAlignment: (String, Int) Positions the text inside the field bounds.

3.4 Progress Bar



CurrentValue: (String, Int, Float, Enum, Date, Time) Current value of the progress bar.

MinValue: (String, Int, Float, Date, Time) Minimum value that the progress bar can go.

Maxvalue: (String, Int, Float, Date, Time) Maximum value that the progress bar can go.

TextXPosition: (String, Int, Float) X position of the text relative to the progress bar.

Widgets

TextYPosition: (String, Int, Float) Y position of the text relative to the progress bar.

BackgroundColour: (String) Background colour.

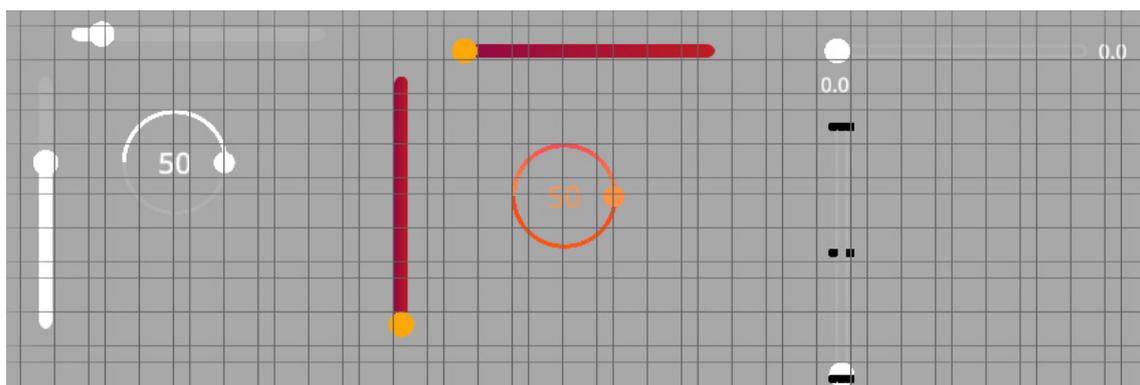
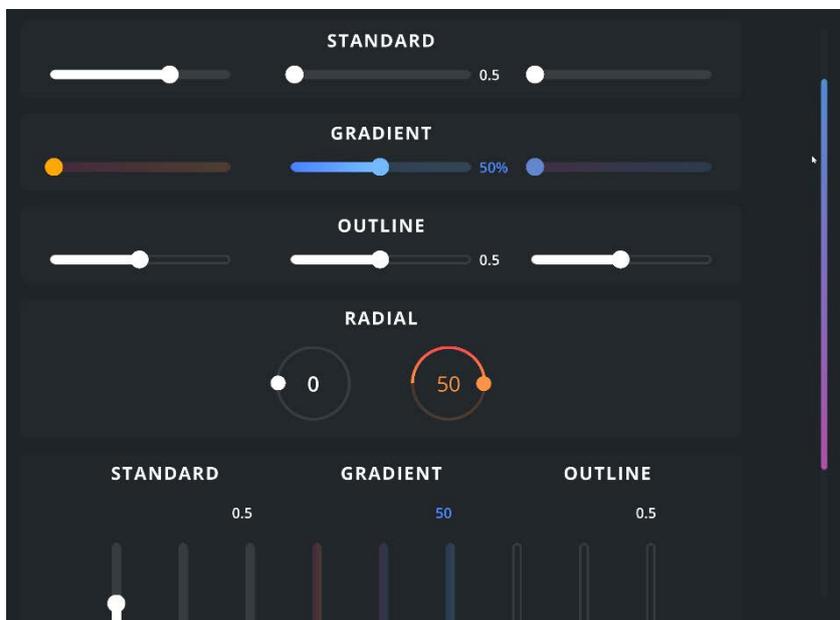
LoadingColour: (String) Colour of the loading bar.

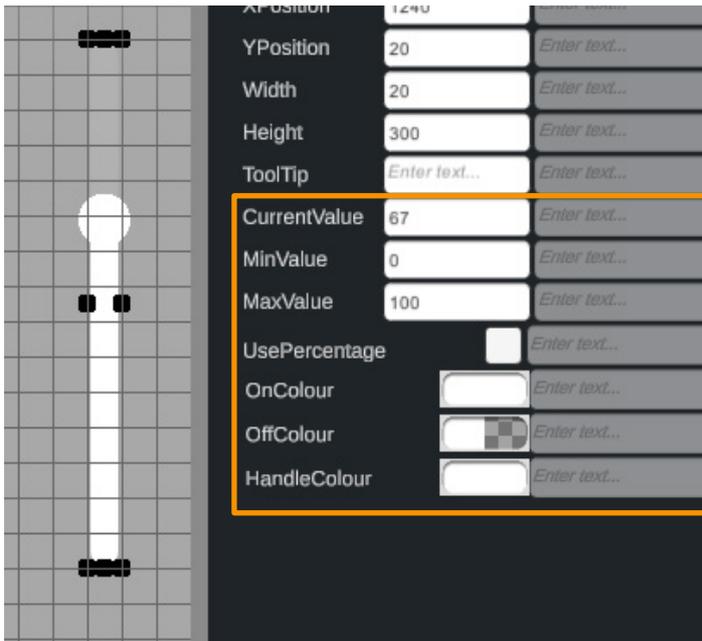
TextColour: (String) Colour of the loading value text.

TextSize: (String, Int, Float) Size of the loading value text.

3.5 Sliders

There are 3 different types of sliders: horizontal, vertical and radial, with style options: standard, outline and gradient.





CurrentValue: (String, Int, Float, Enum, Date, Time) Current value of the slider (Int).

MinValue: (String, Int, Float, Date, Time) Minimum value to which the slider can go.

Maxvalue: (String, Int, Float, Date, Time) Maximum value to which the slider can go.

UsePercentage: (String, Int) If the slider has text, either a percentage or the current value can be shown.

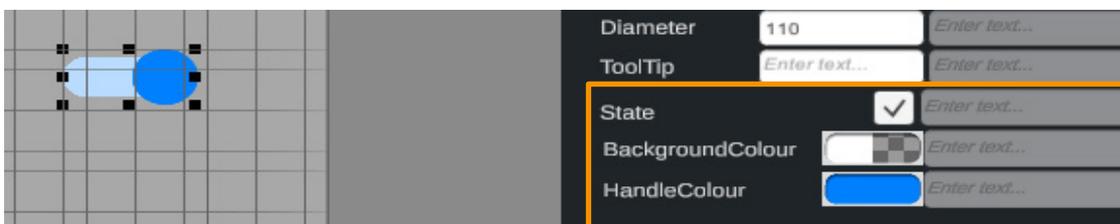
Oncolour: (String) Colour for 'value' part of the slider.

OffColour: (String) Colour for 'available' part of the slider.

HandleColour: (String) Colour of the handle of the slider.

3.6 Toggle (On/Off)

3.6.1 Switch

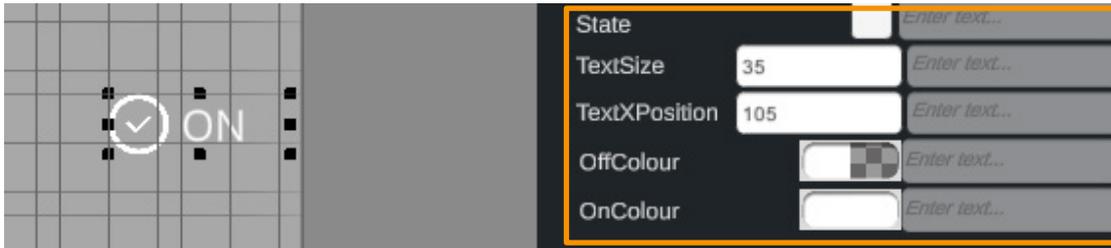


State: (String, Int) Current state of the toggle, checked when on.

BackgroundColour: (String) Background colour of the switch.

HandleColour: (String) Colour of the toggle handle.

3.6.2 Toggle



TextSize: (String, Int, Float) Size of the 'on' and 'off' text.

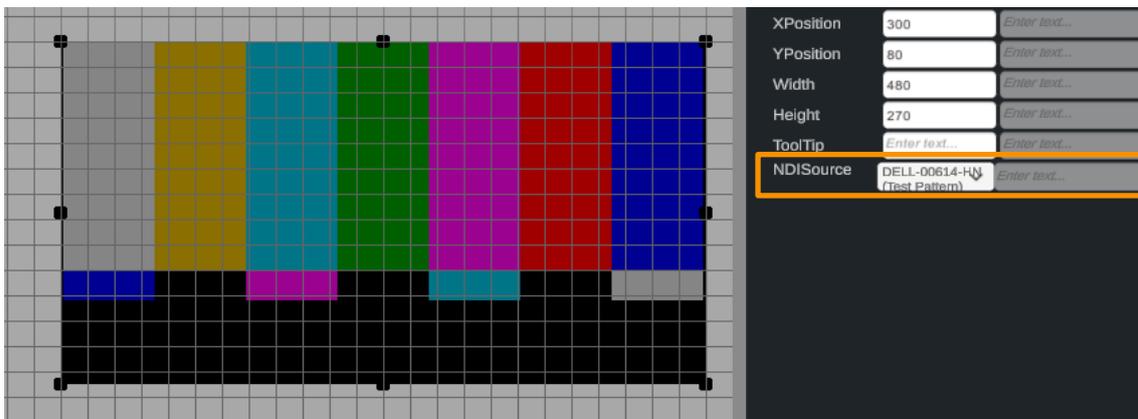
TextXPosition: (String, Int, Float) Position of the text relative to the image. +ve moves the text right, -ve moves it left.

Offcolour: (String) Toggle colour when off.

OnColour: (String) Toggle colour when on.

3.7 Custom

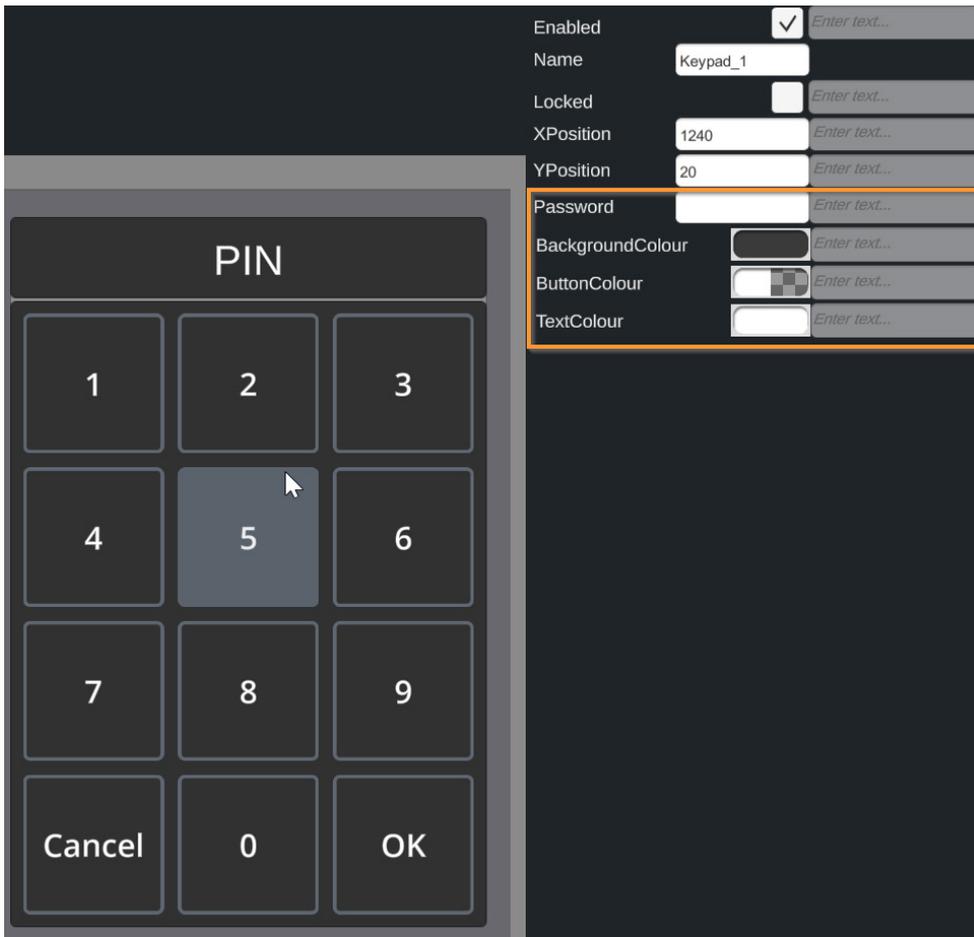
3.7.1 NDI



NDISource: (String) The source of the NDI stream.

3.7.2 Keypad

The Keypad is used with [Preferences > User Levels](#). For example, if you have three user levels then you can type three semicolon-delimited passwords into the password attribute. The index (1;2;3) corresponds to the User Levels configured in the settings menu. When a user enters a valid password, the Marquee project assumes the corresponding level of user access, drawing/hiding/ enabling/disabling widgets throughout the project.



Password: (String) Semicolon delimited list for each password that corresponds to the same index of the user level.

ButtonColour: (String) Colour for the buttons.

TextColour: (String) Colour of the button text.

Example:

User Levels = Admin, Technician, User

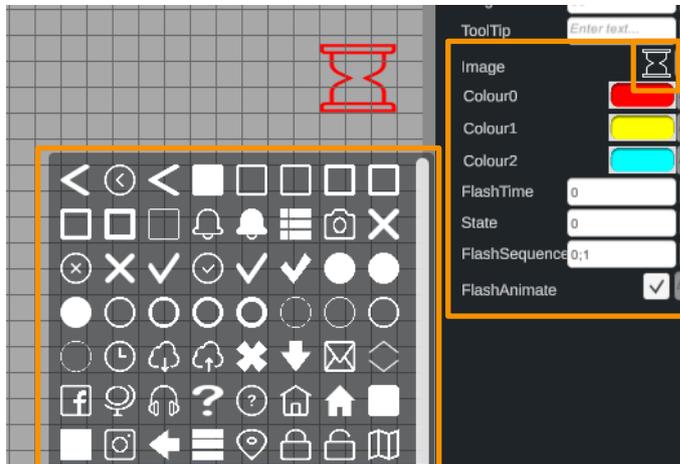
When entered in this order, Admin = Level 1, Technician = Level 2, User=Level 3

Keypad Password (Level) = 789;654;123;

gives PIN access to: Admin; Technician; User in the same order.

3.8 LED

LEDs are indicators based on the same image-character palette as available for Buttons. An LED can be given up to 3 colours.



Colour0: (String) First colour that the LED can be.

Colour1: (String) Second colour that the LED can be.

Colour2: (String) Third colour that the LED can be.

FlashTime: (String, Int, Float) When set to a non-zero value, this represents the time in seconds it takes to switch between or crossfade between the states in the sequence specified in the FlashSequence. (see FlashAnimate, below)

State: (String, Int) Current state of the LED.

FlashSequence: (String) Semicolon separated sequence of states for the LED to flash in. 0;1 by default

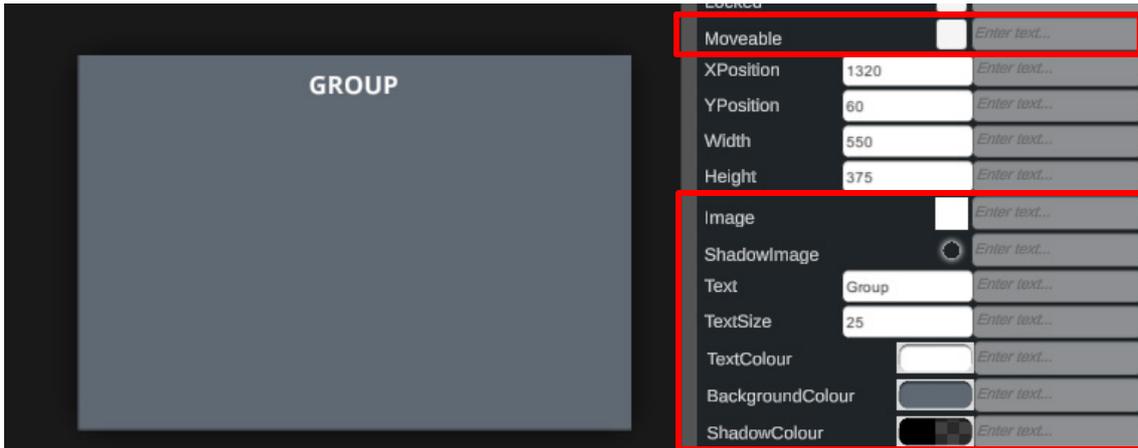
FlashAnimate: (String, Int) Animation of the flashing: either switch through the sequence (off) or blend through colours over the FlashTime period (on).

3.9 Groups

Groups are used to bind widgets together. Add and set the attributes to a group, then select it to add widgets to the group, not the page. Widgets move with the group, remain in the bounds of the group, and are deleted with the group. Group visibility governs its widget's visibility.

3.9.1 Standard Group

In a standard group, child widgets can be moved freely and placed anywhere inside the group.



Moveable: (String, Int) Allow the group to be moved around the screen when in run mode.

Image: (String) Image of the group.

ShadowImage: (String) Image of the shadow behind the group.

Text: (String, Int, Float) Text for the group title.

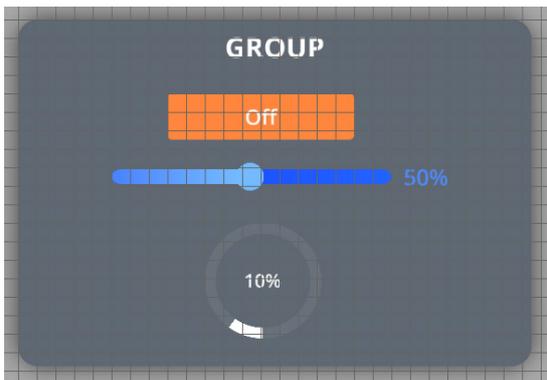
TextSize: (String, Int, Float) Size of the group title text.

TextColour: (String) Colour of the group title text.

BackgroundColour: (String) Colour of the group background.

ShadowColour: (String) Colour of the shadow surrounding the group.

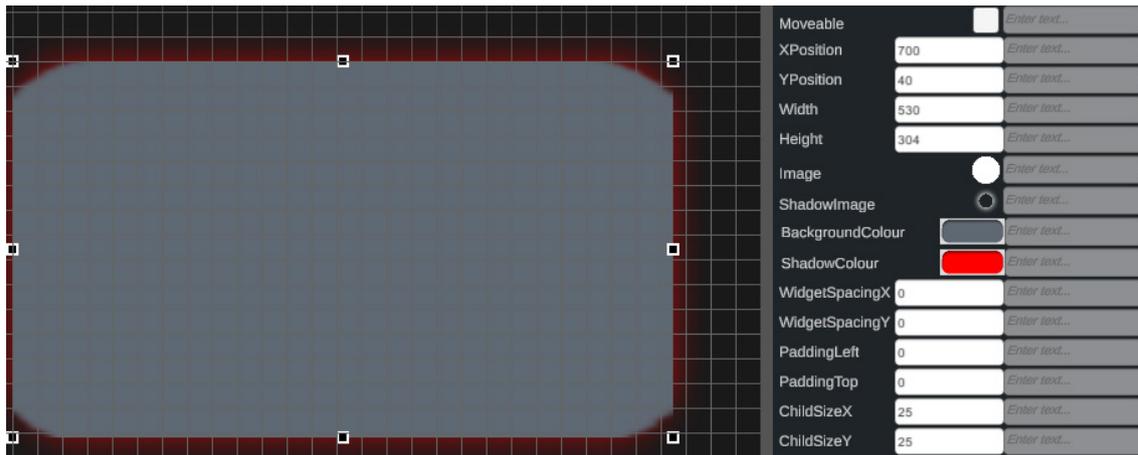
Example:



3.9.2 Grid Group

In a grid group, a widget is placed and sized according to the group's spacing sizing and padding attributes. Set the group attributes first, then select the group before adding grid-spaced widgets. Widgets can be resized, but the origin of each remains at its grid position.

Widgets



Moveable: (String, Int) Controls if the group in run mode is able to be moved around the screen.

Image: (String) Image of the group.

ShadowImage: (String) Image of the shadow behind the group.

BackgroundColour: (String) Colour of the background.

ShadowColour: (String) Colour of the shadow.

WidgetSpacingX: (String, Int, Float) The space in between each widget along the x-axis.

WidgetSpacingY: (String, Int, Float) The spacing of each widget along the y-axis.

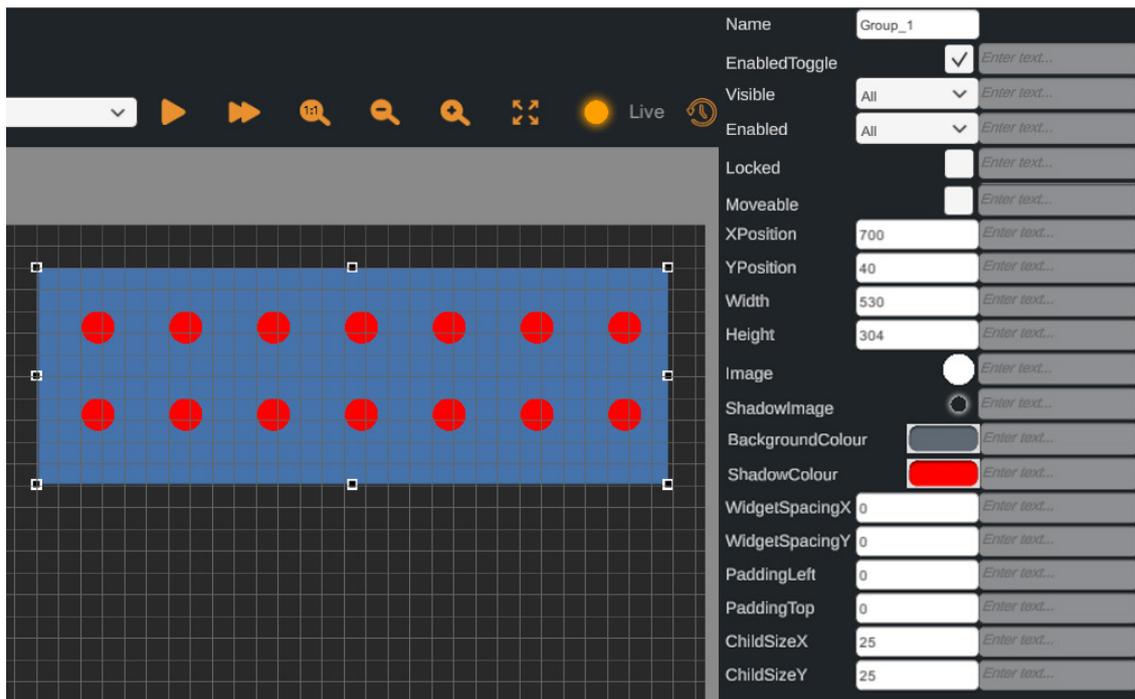
PaddingLeft: (String, Int, Float) Spacing from the left edge of the group boundary.

PaddingTop: (String, Int, Float) Spacing from the top edge of the group boundary.

ChildSizeX: (String, Int, Float) Horizontal size of all individual widgets in the group.

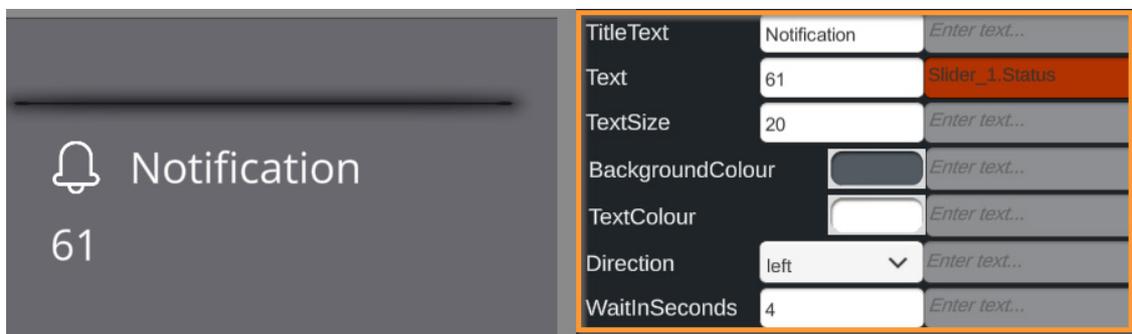
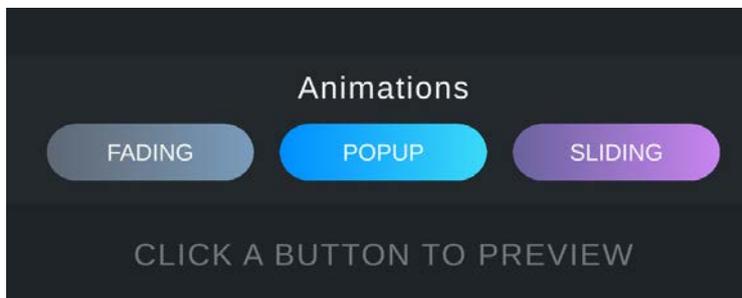
ChildSizeY: (String, Int, Float) Vertical size of all individual widgets in the group.

Example of auto-positioning and sizing of multiple LEDs added to a grid group:



3.10 Notifications

Notifications provide an animated response to value changes. There are three animation types:



TitleText: (String, Int, Float) Text for the notification title.

Text: (String, Int, Float) The text for the notification message.

Widgets

TextSize: (String, Int, Float) Size of all the text.

BackgroundColour: (String) Colour of notification tile.

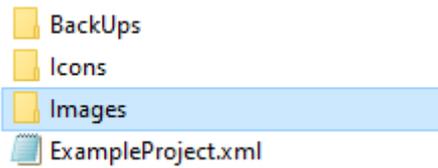
TextColour: (String) Colour of notification and title text.

WaitInSeconds: (String, Int, Float) How long the notification is displayed.

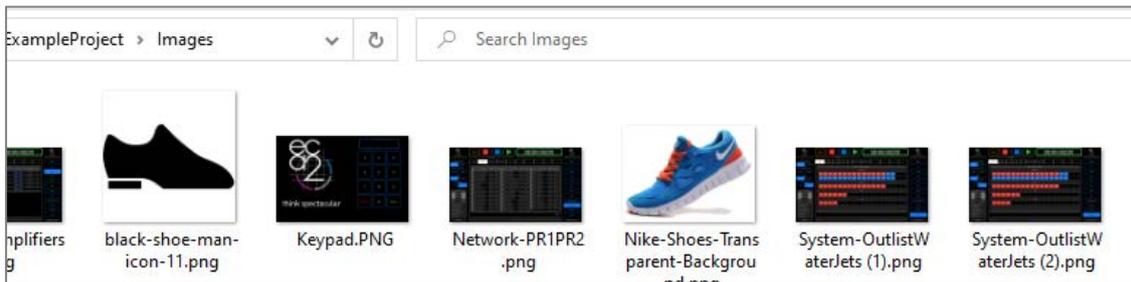
Direction: (String) Direction of animation for Sliding type Notifications: either 'left' or 'right'.

3.11 Gallery

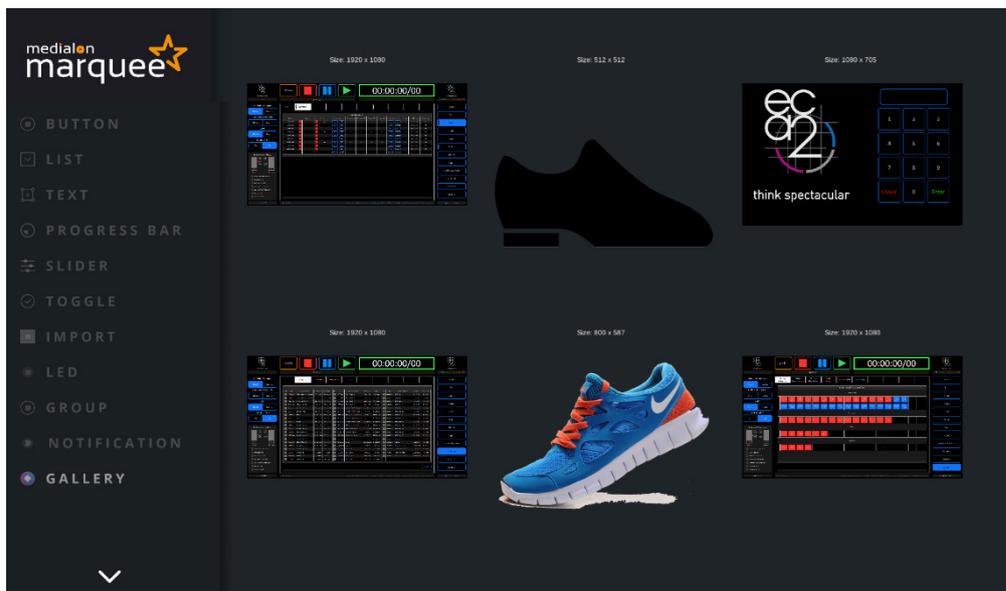
Images can be added to a Marquee page. The project folder contains an images folder. Image files from the Images folder appear in the Widget Picker's Gallery tab:



in the folder in file explorer:



and in the widget picker Gallery:



4 Connecting to Manager

Marquee can connect to one instance of Medialon Manager, but you can connect to multiple shared groups inside Manager. Inside Marquee you can manipulate variables, tasks and devices.

4.1 Set Up

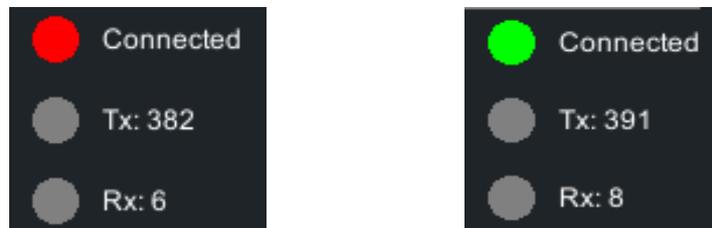
4.1.1 Connection

To Connect to Manager, first enter the correct IP and Port number in *Edit > Preferences*. Enter an IP and Port:

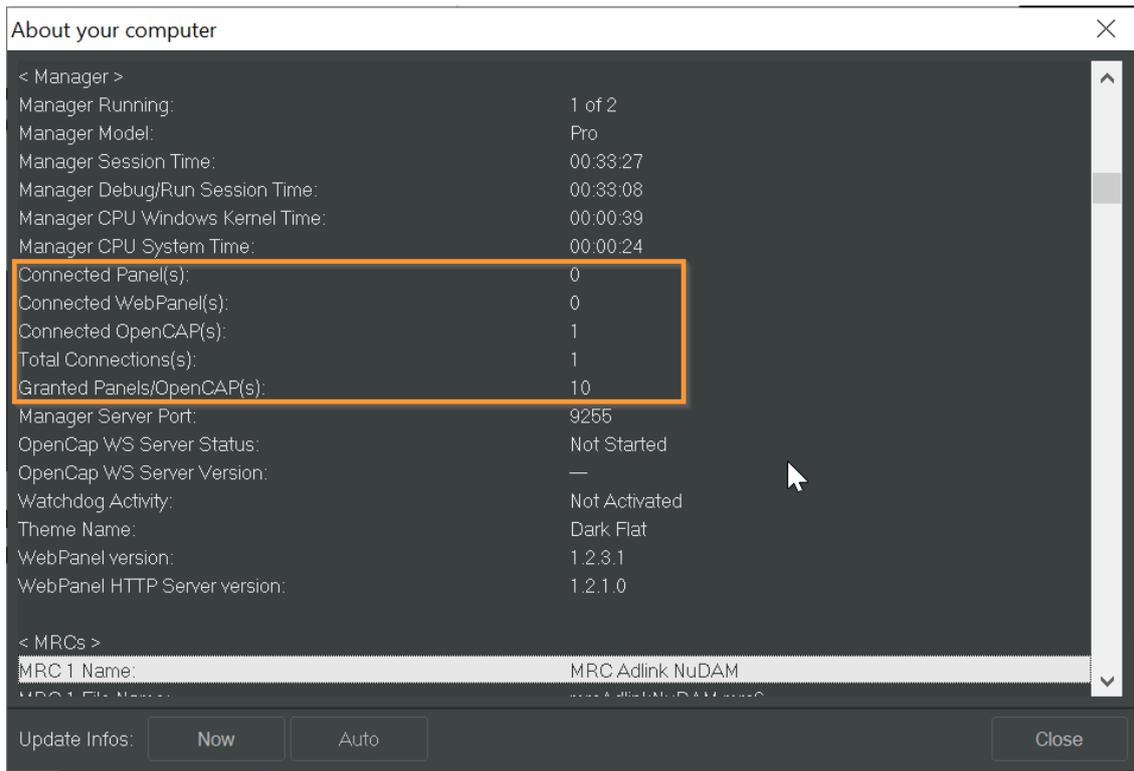


The IP address needs to be the IP of the computer on which Manager is running, and the port default is that of Manager. To find Manager's port in Manager, go to *Edit > Preferences > Network* and the port is shown at the top.

Enter these, and you should see the 'Connected' LED go from Red to Green:



If the 'Connected' LED goes orange it means Marquee has connected at socket level but not API level. This might indicate that your Manager/Showmaster does not have enough available OpenCAP / WebPanel connections.



You can check in Medialon Manager's *Help > About Your Computer* dialog box in the <Manager> section to see how many WebPanel, Manager Panel, and OpenCAP API connections there are vs how many are granted for your show control product.

4.1.2 Set up Shared Groups

For Marquee to be able to access Manager's variables, they need to be in a Shared Group. You can have multiple Shared Groups, but at least one is needed.

In Marquee go to the *Edit > Preferences > Manager* Tab and you will see a list of shared groups which by default contains one. Use the plus/minus to add/remove groups.

Example:



In my **Manager** project I have two shared groups. One for variables (Group1) and one for tasks (Group2).

In **Marquee**, I add both of these to my Shared Group list, as I want to be able to access both variables and tasks.

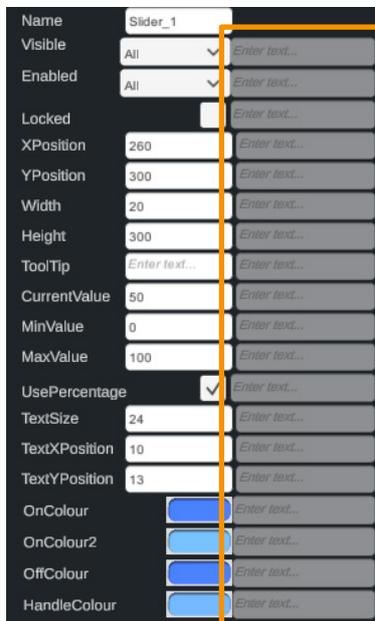
4.2 Connect Widgets to Shared Variables/ Tasks

You can connect any widget's attributes to Manager's shared variables and tasks through the Inspector. This is a two-way dynamic communication: when you change the value of a widget in Marquee (for instance a Slider), the linked shared variable updates in Manager and vice versa.

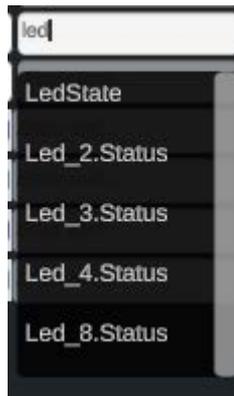
Note: The Name attribute cannot be connected to a shared variable, as it is a unique identifier in the Marquee project.

4.2.1 Manager Variables

To connect widget attributes to shared variables, first open the Inspector by clicking on the widget. The greyed-out input fields for each attribute are where the Shared Variables go.

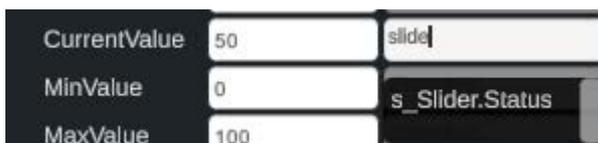


These input fields are also lists. When you start typing they will be filtered; select the variable/ task you want:



Example: connect a Marquee Slider to a Manager Slider:

- Drag Manager's **Slider.Status** value into the Shared Group so Marquee can access it.
- Open Marquee's Slider in the Inspector and locate the attribute called 'CurrentValue'.
- In the shared variable input field, start typing the name of Manager's shared slider and the list will filter until you see 's_Slider.Status' as a shared variable. Select this variable from the list.



- The Marquee slider and the Manager slider are now linked. With Marquee in Live mode (F5) and Manager running, the sliders move together.

Shared Variable Accepted?

	Accepted
	No shared variable entered
	Variable not contained in Shared Groups
	Unsupported type for the attribute (ignored)

Supported types

Different attributes can be connected to different Manager variables. The type (string, int, float/real, enum, time and date) must correspond. The variable types denoted in this documentation for the widget attributes explains the type of Manager variables that can be linked to that attribute.

Connecting colour variables to Manager

Colour is formatted in RGBA (0-255;0-255;0-255;0-225). Create a string in Manager of '255;0;0;255' (i.e. red, full opacity) and any colour attribute using that shared string variable as its shared variable will display as red.

Dynamically change icon images

Any .png or .jpg images saved in your Image/Icon folder can be assigned as icons by using a string variable in Manager. This should contain the name of the image to be assigned as an icon (e.g. 'MyImage.png'). Then connect the Icon shared variable to the string variable in Manager. You can change the value of the string during runtime and the icon's image updates.

Connecting Widgets to Time

Requirements:

- Marquee: 1 slider and 3 text widgets.
- Manager: 3 time variables (Start, End, Current)

Steps:

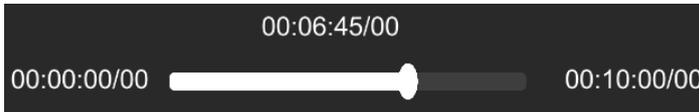
- Attach the left InputField text attribute to Start
- Attach the middle InputField text attribute to Current
- Attach the right InputField text attribute to End
- Slider.CurrentValue shared var = Current
- Slider.MinValue shared var = Start
- Slider.MaxValue shared var = End

Connecting to Manager

In Manager:

	StartTime	00:00:00/00 - 100 fps
	EndTime	00:10:00/00 - 100 fps
	CurrentTime	00:06:45/00 - 100 fps

In Marquee:



4.2.2 Manager Tasks

You can Start, Pause or Stop tasks from Marquee by connecting button actions to tasks.

Open up a button widget in Inspector and find the 'Action Down' or 'Action Up' attribute, and select either 'Start', 'Pause', or 'Stop' in the dropdown. Then for the Shared Variable, enter the task you want to manipulate with the button. The button action and task are now linked.

4.3 Marquee Syntax for Single Instruction

You can also send ad-hoc commands to Marquee from Manager using a shared string variable named 'Marquee'.

This is done by using the syntax:

`<Widget Name>.<Key>=<Value>`

The Key has to match the attribute key in Marquee's Inspector. These vary from widget to widget. To find them, add a widget and open its Inspector.

Example:

'Slider_2.CurrentValue=55'

'Slider_2.Colour=55,0,125,255'

Widget Name = Slider_2

Widget Name = Slider_2

Key = Status

Key = Colour

Value = 55

Value = 55,0,125,255

Wild cards '*' and '?' are also supported.

4.3.1 Internal Attribute Communication



To control an attribute of the same or a different widget within the Marquee project itself, use this syntax:

Connecting to Manager

Tilde(~)<Widget Name>.<Attribute Key>

For example the string of '~InputField_0.Text' in the shared variable of 'CurrentValue' in a slider, will set the Text attribute of the InputField to the slider value.



medialon

BY 7THSENSE

E: info@7thsense.one
W: [7thsense.one](https://www.7thsense.one)

7thSense Design Ltd

2 The Courtyard Shoreham Road
Upper Beeding
Steyning
West Sussex
BN44 3TN
UK

T: +44 (0) 1903 812299

7thSense LLC, Michigan

332 E Lincoln Ave
Suite 100
Royal Oak, MI 48067
USA

T: +1 248 599 2717

7thSense LLC, Orlando

4207 Vineland Rd
Suite M1
Orlando, FL 32811
USA

T: +1 407 505 5200
