

ORBIT FOR MULTIVIEWERS

ON-SCREEN VIDEO WALL MONITORING AND GRAPHICS

User Manual

Issue 1 Rev 3

2019-05-28

Patent Information

This product may be protected by one or more patents.

For further information, please visit: www.grassvalley.com/patents/

Copyright and Trademark Notice

Copyright © 2019, Grass Valley Canada. All rights reserved.

Belden, Belden Sending All The Right Signals, and the Belden logo are trademarks or registered trademarks of Belden Inc. or its affiliated companies in the United States and other jurisdictions. Grass Valley, Orbit are trademarks or registered trademarks of Grass Valley Canada. Belden Inc., Grass Valley Canada, and other parties may also have trademark rights in other terms used herein.

Terms and Conditions

Please read the following terms and conditions carefully. By using Orbit documentation, you agree to the following terms and conditions.

Grass Valley hereby grants permission and license to owners of Orbit to use their product manuals for their own internal business use. Manuals for Grass Valley products may not be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose unless specifically authorized in writing by Grass Valley.

A Grass Valley manual may have been revised to reflect changes made to the product during its manufacturing life. Thus, different versions of a manual may exist for any given product. Care should be taken to ensure that one obtains the proper manual version for a specific product serial number.

Information in this document is subject to change without notice and does not represent a commitment on the part of Grass Valley.

Warranty information is available from the Legal Terms and Conditions section of Grass Valley's website (www.grassvalley.com).

| | |
|-------------|------------------------------------|
| Title | Orbit for Multiviewers User Manual |
| Part Number | Issue 1 Rev 3 |
| Revision | 2019-05-28, 10:01 |

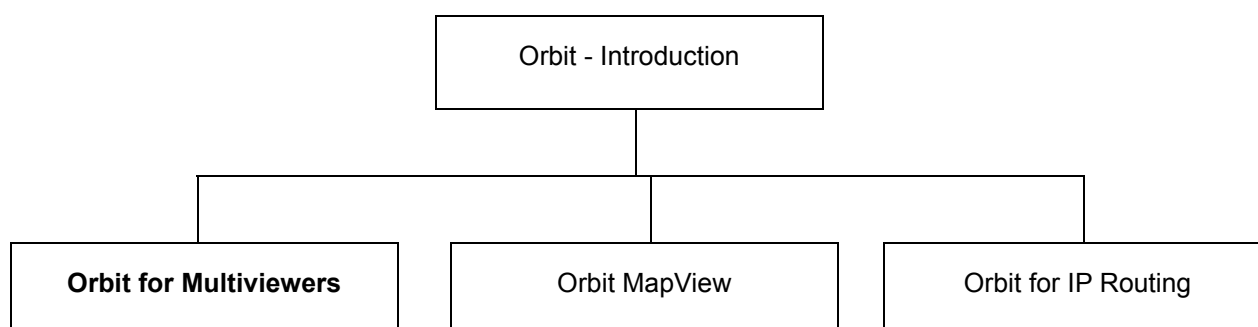
About this Manual

This Orbit user manual describes how to use Orbit software for multiviewer applications. Introductory information is found in the 'Orbit - Introduction User Manual'.

The following Orbit manuals are related documents:

| | Document Title | Description |
|--------------|---|---|
| | Orbit - Introduction User Manual | A general introduction to the Orbit software and its application. |
| This manual: | Orbit for Multiviewers User Manual | Describes multiviewer-specific features of Orbit. |
| | Orbit for IP Routing User Manual | Describes IP routing features of Orbit. |
| | Orbit MapView User Manual | Describes Orbit MapView soft screens. |

Table 1 Related Orbit Documents



toc

Table of Contents (Summary)

| | |
|--|------------|
| Patent Information | ii |
| Copyright and Trademark Notice | ii |
| About this Manual | iii |
| 1 Multiviewer Projects | 1 |
| 2 Wall Editor | 6 |
| 3 Tile Editor - Tiles and Widgets | 73 |
| 4 Theme Editor - Widget Styles | 112 |
| 5 Widget Behaviours and Bindings | 130 |
| 6 Groups of Widgets | 146 |
| 7 Project Variables | 167 |
| 8 Example - Basic Multiviewer Video Wall | 188 |
| 9 Examples - UMDs and Border Tally | 228 |
| 10 Examples - Other On-screen Functionality | 285 |
| Appendix A Terminology | 318 |
| Appendix B List of Widgets | 321 |
| Appendix C List of Behaviours and Bindings | 332 |
| Grass Valley Technical Support | 372 |
| Corporate Head Office | 372 |

toc

Table of Contents

| | |
|--|----------|
| Patent Information | ii |
| Copyright and Trademark Notice | ii |
| About this Manual | iii |
| 1 Multiviewer Projects. | 1 |
| 1.1 Introduction | 2 |
| 1.2 Multiviewer Project Name | 3 |
| 1.2.1 Renaming a Project | 3 |
| 1.3 Setting up Multiviewer Device Details in a Project | 4 |
| 1.4 Editor Screens and Editing a Project using this Manual | 5 |
| 2 Wall Editor | 6 |
| 2.1 Introduction | 7 |
| 2.2 How to Open and Create Walls | 8 |
| 2.2.1 Create a New Video Wall using the Walls Icon | 9 |
| 2.2.2 'Wall file name' and the 'Wall Name' | 11 |
| 2.2.3 Create a New Wall via the Project View | 11 |
| 2.2.4 Open an Existing Video Wall | 12 |
| 2.2.5 Open Existing Video Wall via Project View | 12 |
| 2.3 Introduction to The Wall Editor Screen | 14 |
| 2.3.1 Wall Editor Tool Bar | 15 |
| 2.4 How to Assign Multiviewer Outputs to a Wall | 16 |
| 2.4.1 Assign Outputs Editor Screen | 16 |
| 2.4.2 Outputs Banner | 18 |
| 2.4.3 Assigning Display Outputs | 19 |
| 2.4.4 Display Output Properties | 21 |
| 2.4.5 Display Wall Properties | 22 |
| 2.4.6 Selecting a Color Value | 24 |
| 2.4.7 Display Output Context Menu | 25 |
| 2.5 Tile Arrangement on a Wall | 28 |
| 2.5.1 The Tile Grid and Fine Grid Modes | 28 |
| 2.6 Wall Editor - Tile Grid Mode | 31 |

| | |
|---|----|
| 2.6.1 Add Tile Grid Example | 33 |
| 2.6.2 Adding a Tile Grid - Using "Output:" drop-down box | 34 |
| 2.6.3 Adding a Tile Grid - using Grid Rows and Columns..... | 35 |
| 2.6.4 Adding a Tile Grid and Adjusting Tile Spacing | 37 |
| 2.6.5 Resize a Tile in Tile Grid Mode..... | 37 |
| 2.6.6 Add Current Grid, '1+20 Tile' Grid..... | 40 |
| 2.6.7 Reposition a Tile in Tile Grid Mode | 40 |
| 2.6.8 Tile Grid Mode - The Clear and Fill Output Controls | 41 |
| 2.6.9 Tile Grid Mode - Select, Delete, Resize and Fill | 43 |
| 2.7 Wall Editor - Fine Grid Mode | 46 |
| 2.7.1 Arranging Tiles..... | 48 |
| 2.8 Fine Adjustment of Tiles on a Wall | 51 |
| 2.8.1 Fine Position and Size Adjustment | 52 |
| 2.8.2 Layering (Z order) of Tiles..... | 53 |
| 2.8.3 Fine Adjustment Example:..... | 54 |
| 2.9 Add a Custom Tile to Wall..... | 55 |
| 2.10 Wall Tile Properties..... | 57 |
| 2.11 Assign Inputs to a Wall | 64 |
| 2.11.1 Automatic Assignment of Inputs | 64 |
| 2.11.2 Manual assignment of video inputs..... | 67 |
| 2.12 Wall Layouts | 68 |
| 2.12.1 How to Create a new Wall Layout | 68 |
| 2.12.2 Switch Wall Layouts in the Orbit Wall Editor | 70 |
| 2.12.3 Switch Wall Layouts in Orbit in Run Mode (MV-800-DT License) | 71 |
| 2.12.4 Switch Wall Layouts on a Multiviewer Unit | 71 |

3 Tile Editor - Tiles and Widgets 73

| | |
|---|-----|
| 3.1 Introduction | 74 |
| 3.2 Terminology | 75 |
| 3.3 Opening a Tile Design in the Tile Editor | 76 |
| 3.3.1 Creating a New Tile..... | 76 |
| 3.3.2 Creating a New Tile Design from an Auto-Tile | 80 |
| 3.3.3 Opening an Existing Tile Design | 81 |
| 3.3.4 Opening Tiles using the Project View..... | 82 |
| 3.4 Tile Editor Screen..... | 84 |
| 3.4.1 Tile Editor Tab | 85 |
| 3.4.2 Main Tool Bar items | 86 |
| 3.4.3 Properties Box | 86 |
| 3.4.4 Selecting Graphical Objects when Editing in Tile Editor | 87 |
| 3.5 Placing Widgets..... | 88 |
| 3.5.1 Widgets Tool Bar | 88 |
| 3.5.2 Add a Widget to a Tile or Schematic..... | 88 |
| 3.5.3 Reposition/Resize Widgets | 89 |
| 3.6 Schematic 'Right-Click' Context Menus..... | 92 |
| 3.7 Tile Editor - Advanced Tool Bar..... | 99 |
| 3.8 Auto Fill Facility | 101 |
| 3.8.1 Example - Labeling Widgets in a Numbered Sequence..... | 102 |
| 3.9 Auto-Tiles for Multiviewers | 105 |

| | |
|---|------------|
| 3.9.1 Video Auto-Tiles - 3G/HD and 4K..... | 107 |
| 3.9.2 Video Auto-Tile - Style Properties | 108 |
| 3.9.3 Video Auto-Tile - Properties..... | 109 |
| 3.9.4 Video Auto-Tile - Appearances..... | 110 |
| 3.9.5 Clock Auto-Tiles..... | 111 |
| 4 Theme Editor - Widget Styles | 112 |
| 4.1 Open a Theme in the Theme Editor | 113 |
| 4.1.1 Creating a new Theme | 113 |
| 4.1.2 Opening an Existing Theme | 115 |
| 4.1.3 Theme Editor from the Project View..... | 115 |
| 4.1.4 Setting the Current Theme from the Project View..... | 117 |
| 4.2 Theme Editor Screen | 118 |
| 4.2.1 Theme Editor Tab | 119 |
| 4.3 Widget Category Tabs | 120 |
| 4.4 Widget Properties in Theme Editor..... | 122 |
| 4.5 'Default.theme' in a New Project | 123 |
| 4.6 Create a Widget Style for a Theme | 124 |
| 4.7 Using Widget Styles | 127 |
| 4.7.1 Default to Default Style: | 129 |
| 5 Widget Behaviours and Bindings | 130 |
| 5.1 Introduction | 131 |
| 5.1.1 Behaviour and Binding Example Use-Cases | 131 |
| 5.2 Behaviours and Bindings Graphical Editor..... | 134 |
| 5.2.1 Behaviours and Bindings Editor Screen | 135 |
| 5.2.2 Behaviours and Bindings Vertical Tabs | 137 |
| 5.2.3 Behaviour and Bindings Editor - Tool Bar | 138 |
| 5.2.4 Behaviour and Bindings Editor - Block Diagram Area | 138 |
| 5.3 Widget Property Types | 142 |
| 5.3.1 Using Orbit Variables with Strings..... | 143 |
| 5.3.2 Enumerations (Named Value) | 143 |
| 5.3.3 Setting Color Values..... | 143 |
| 5.4 Video Auto-Tiles..... | 144 |
| 6 Groups of Widgets | 146 |
| 6.1 Widget Groups..... | 147 |
| 6.1.1 Create a Group..... | 147 |
| 6.1.2 Undoing a Group (Ungroup)..... | 149 |
| 6.1.3 Selecting and Moving a Group..... | 150 |
| 6.1.4 Removing a Widget from a Group..... | 151 |
| 6.1.5 Adding a Widget to a Group | 152 |
| 6.1.6 Resizing a Group and Group Layout..... | 153 |
| 6.2 Properties of a Group | 154 |
| 6.2.1 Layout Property Examples..... | 156 |
| 6.2.2 Adding / Moving a Widget and Automatic Repositioning of Group Items... .. | 158 |
| 6.3 Group Layout | 160 |
| 6.3.1 Spacers and Group Layouts..... | 160 |
| 6.3.2 Widget Layout Item Properties..... | 163 |
| 6.4 Group Layout Example - Soft Panel | 165 |
| 7 Project Variables | 167 |
| 7.1 Introduction to Project Variables..... | 168 |

| | |
|---|------------|
| 7.1.1 Application Examples for Variables. | 168 |
| 7.1.2 How to Create a User Project Variable | 169 |
| 7.1.3 Using a User Project Variable | 170 |
| 7.2 Tile Variables | 175 |
| 7.2.1 Right-click 'Variables...' Menu Item. | 175 |
| 7.2.2 Set Variables Icon and User Tile Variables | 178 |
| 7.2.3 User Tile Variable and Set Variables Example: | 181 |
| 7.2.4 'Create Variable from Property' Example | 184 |
| 8 Example - Basic Multiviewer Video Wall. | 188 |
| 8.1 Default Project | 189 |
| 8.1.1 Pull Default Project from Multiviewer (Connected Multiviewer Project) | 190 |
| 8.1.2 Make a Simple Change to the Default Project | 191 |
| 8.1.3 Push the Project to the Multiviewer | 194 |
| 8.1.4 Exercising the alarm | 195 |
| 8.2 Orbit Initial Screen | 196 |
| 8.3 Creating a New Project | 197 |
| 8.3.1 New Project - Multiviewer project | 197 |
| 8.3.2 New Project - Connected Multiviewer Project | 200 |
| 8.4 Opening an Existing Project | 203 |
| 8.5 Opening a Recent Project | 204 |
| 8.6 Orbit Project Home Screen | 205 |
| 8.6.1 Orbit Project Status Bar | 206 |
| 8.7 Making a Basic Video Wall Layout | 207 |
| 8.7.1 Create a New Multiviewer Project | 207 |
| 8.7.2 Create the Video Wall | 209 |
| 8.7.3 Import a Logo Image into the Project | 213 |
| 8.7.4 Create Custom Tiles | 214 |
| 8.7.5 Add Custom Tiles to the Wall | 216 |
| 8.7.6 Assign Multiviewer Inputs to Wall Video Tiles | 218 |
| 8.8 Push Project to the Multiviewer | 220 |
| 8.8.1 Before Pushing a Project | 220 |
| 8.8.2 Check Project Name | 220 |
| 8.8.3 Push the Project to the Multiviewer | 220 |
| 8.9 Wall Layouts | 221 |
| 8.10 Themes | 223 |
| 8.10.1 Baseline Default Theme | 223 |
| 8.10.2 Creating Themes | 223 |
| 8.10.3 Applying Themes to a Wall | 225 |
| 8.10.4 Applying a Style to a Widget | 226 |
| 8.10.5 Locally Modifying the Style of a Widget | 226 |
| 8.10.6 Default Style | 227 |
| 9 Examples - UMDs and Border Tally. | 228 |
| 9.1 Using a Built-in UMD Widget | 230 |
| 9.1.1 Set Up The Behaviour | 231 |
| 9.1.2 Set Up the Binding | 233 |
| 9.2 Do-It-Yourself UMD (Lamps and a Label) | 236 |
| 9.2.1 Configuring Behaviours and Bindings | 237 |
| 9.2.2 Lamp Widgets | 241 |
| 9.2.3 Label Widget - Border Color | 246 |
| 9.2.4 Label Widget - Font Color | 249 |
| 9.2.5 Label Widget - Caption Text | 251 |
| 9.2.6 D-I-Y UMD Behaviour and Bindings Recap | 254 |
| 9.3 Another Do-It-Yourself UMD (Shape Fill and Images) | 255 |

| | |
|---|------------|
| 9.3.1 Configuring Behaviours and Bindings | 256 |
| 9.3.2 Shape Widget | 258 |
| 9.3.3 Image Widget | 261 |
| 9.3.4 Label Widget | 262 |
| 9.3.5 D-I-Y UMD Behaviour and Bindings Recap | 263 |
| 9.4 Border Tally Example | 264 |
| 9.4.1 Configuring Behaviours and Bindings | 265 |
| 9.4.2 Border Behaviour and Binding | 270 |
| 9.4.3 Label Widget | 273 |
| 9.4.4 Border Tally Behaviour and Bindings Recap | 278 |
| 9.5 Alternative Border Tally | 280 |
| 9.6 Exercise the Tile and Widgets | 283 |
| 10 Examples - Other On-screen Functionality | 285 |
| 10.1 On-screen Notification from RollCall Log Server | 286 |
| 10.1.1 Preliminaries | 287 |
| 10.1.2 Setting up the Lower Label - On-Screen | 288 |
| 10.1.3 Setting Up the Upper Label - Behaviour | 289 |
| 10.1.4 Upper Label - Binding Background Fill Color | 291 |
| 10.1.5 Image Widget Set Up | 293 |
| 10.1.6 Finally | 295 |
| 10.2 Adding a GPI Output to a Tile | 297 |
| 10.2.1 Setting up a Multiviewer's GPIO to be an Output | 298 |
| 10.2.2 Custom Tile and GPIO | 299 |
| 10.2.3 Auto-Tile and GPI Output | 303 |
| 10.3 On-Screen Video Input Status | 306 |
| 10.3.1 Create a Video Tile Style Showing Text Lines | 306 |
| 10.3.2 Set Up the Number of On-Screen Text Status Lines | 308 |
| Appendix A Terminology | 318 |
| A.1 Terminology for Multiviewers | 319 |
| Appendix B List of Widgets | 321 |
| B.1 Widget Tool Bar Items | 322 |
| Appendix C List of Behaviours and Bindings | 332 |
| C.1 List of Behaviours | 333 |
| C.1.1 Alarm Behaviour (Group: Monitoring) | 334 |
| C.1.2 Alarm Mask Behaviour (Group: Monitoring) | 336 |
| C.1.3 Audio Level Behaviour (Group: Multiviewer) | 337 |
| C.1.4 Closed Caption Behaviour (Group: Multiviewer) | 337 |
| C.1.5 Display Details Behaviour (Group: Monitoring) | 339 |
| C.1.6 Local Value Behaviour (Group: General) | 339 |
| C.1.7 Log Field Behaviour (Group: Monitoring) | 340 |
| C.1.8 RollCall Timer Behaviour (Group: Timer) | 342 |
| C.1.9 RollCall v3 Command Behaviour (Group: RollCall) | 343 |
| C.1.10 RollCall+ Command Behaviour (Group: RollCall) | 345 |
| C.1.11 RollTrak Behaviour (Group: RollCall) | 345 |
| C.1.12 Schematic Link Behaviour (Group: General) | 347 |
| C.1.13 TRP Timer Behaviour (Group: Timer) | 348 |
| C.1.14 TSL Behaviour (Group: Multiviewer) | 349 |
| C.1.15 Video Input Behaviour (Group: Multiviewer) | 350 |
| C.2 List of Bindings | 352 |

| | |
|---|-----|
| C.2.1 Audio Bars Binding (Group: Multiviewer) | 353 |
| C.2.2 Checkbox Binding (Group: General) | 355 |
| C.2.3 Direct Binding (Group: General) | 355 |
| C.2.4 Logical Binding (Group: General) | 356 |
| C.2.5 Mapped Value Binding (Group: General)..... | 360 |
| C.2.6 Property Binding (Group: General)..... | 363 |
| C.2.7 Radio Button Binding (Group: General) | 367 |
| C.2.8 RollCall Timer Binding (Group: Timer)..... | 368 |
| C.2.9 RollCall v3 Binding (Group: RollCall) | 369 |
| C.2.10 TRP Timer Binding (Group: Timer) | 369 |
| C.2.11 TSL UMD Binding (Group: Multiviewer)..... | 370 |
| Grass Valley Technical Support | 372 |
| Corporate Head Office | 372 |



1 Multiviewer Projects

Multiviewer Projects

| | |
|--|---|
| Introduction | 2 |
| Multiviewer Project Name | 3 |
| Renaming a Project | 3 |
| Setting up Multiviewer Device Details in a Project | 4 |
| Editor Screens and Editing a Project using this Manual | 5 |

Note:

See the related document 'Orbit - Introduction User Manual' for:

- A full introduction to Orbit for Multiviewers.
- Information about projects: Managing user access to projects; Opening and creating Orbit multiviewer projects.

1.1 Introduction

The layout and style of the multiviewer video walls are designed with the Orbit client software application and held in an Orbit multiviewer project. A new multiviewer project can be created from scratch, or opened from a multiviewer device. If the multiviewer device is a new device, then the default multiviewer project is available.

An Orbit multiviewer plug-in enables the Orbit client software to edit multiviewer projects. Grass Valley's MV-8 series of multiviewers are supported. Multiviewer video walls can be designed and configured.

Video wall designs are pushed to a multiviewer device as Orbit multiviewer projects for deployment.

Multiple wall designs may be generated and the multiviewer project stored on a PC. Different wall designs can then be pushed to the multiviewer for various different Orbit multiviewer applications. Different wall layouts in a project can be changed on the fly.



Figure 1 Orbit for Multiviewers - Video Wall Layout Design to Deployment

1.2 Multiviewer Project Name

The Orbit project name is treated as an identifier by Orbit. Projects should be given unique project names.

Projects with the same name are considered to be the same. Orbit warns the user if there are any differences between a project in Orbit and the same-named project on a connected multiviewer device. Such differences must be resolved.

It is recommended to:

- Use a different and unique project name for each multiviewer.
- Rename a default project that has been pulled from a multiviewer.

1.2.1 Renaming a Project

Once a project is open in Orbit, a project may be renamed by editing the name on the Project Home Screen, if required. (See the 'Orbit - Introduction User Manual'.)

Edit Name control

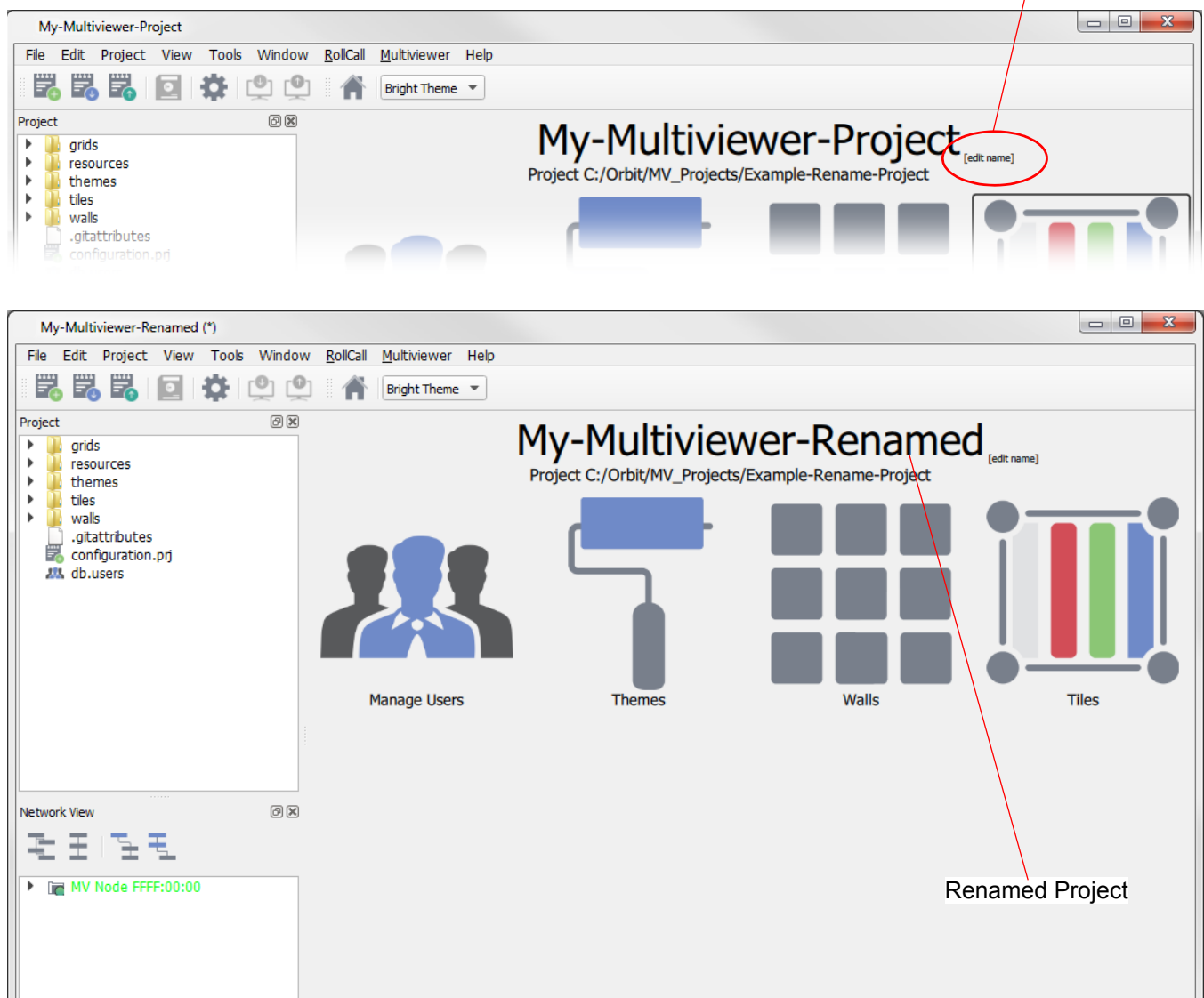


Figure 2 Rename a Project

1.3 Setting up Multiviewer Device Details in a Project

Before a project can be pushed to a multiviewer, the project must be set up with some details about the target multiviewer.

If the project has been created as a new 'connected project', or if it has been pulled from a multiviewer, then multiviewer details will be set up in the project. These multiviewer details will need to be modified to target a different multiviewer.

If the project is new, then multiviewer details must be set up.

To modify or set up multiviewer device details in a multiviewer project:

1. Click **Multiviewer > Properties**. The properties window is displayed, see Figure 3.

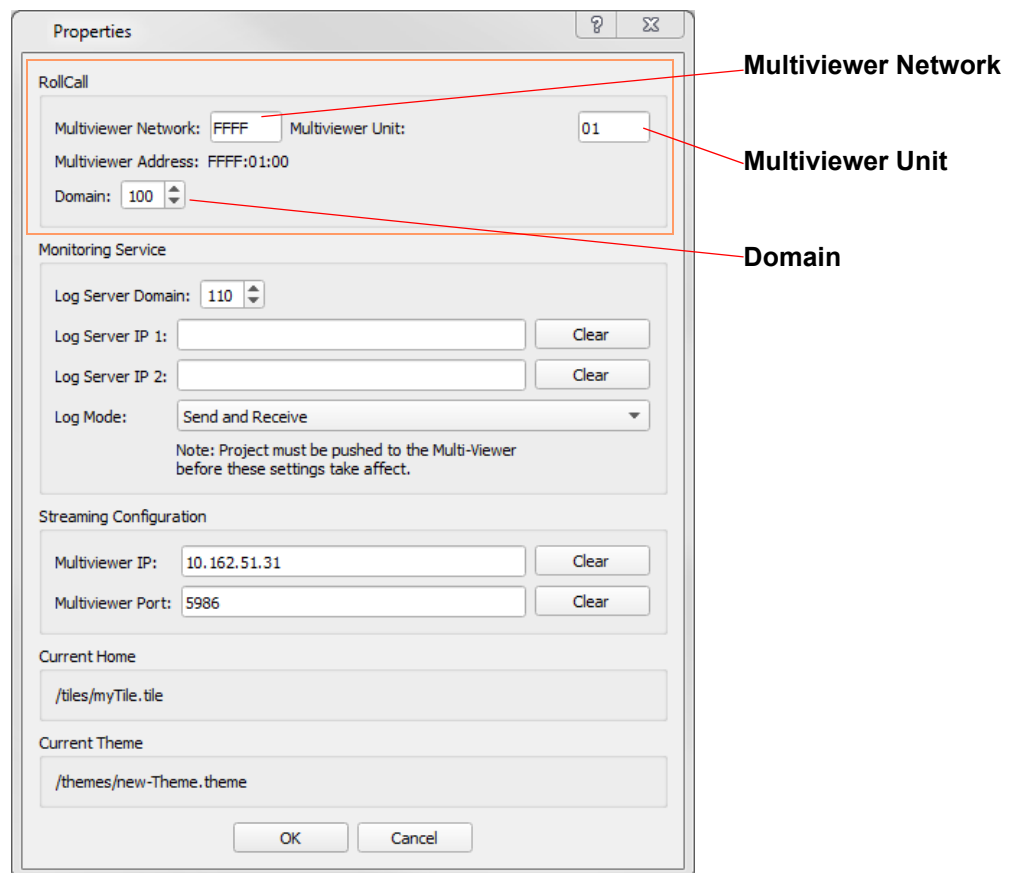


Figure 3 Multiviewer > Properties Window

2. Set the RollCall **Multiviewer Network**, **Multiviewer Unit**, and **Domain** settings to match the target multiviewer.

Note: Ensure the details in the Orbit project match those in the RollCall System-Setup template of the target multiviewer.

3. Click **OK**.

4. Click **Project > Select**. An 'Open remote' dialog is shown, see Figure 4.

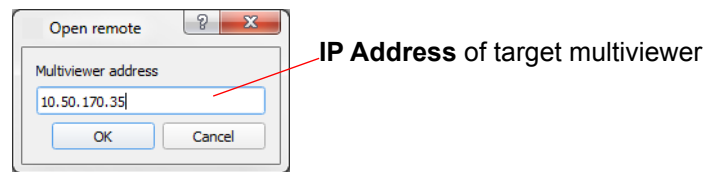


Figure 4 Properties > Select, Open Remote Dialog

5. Enter the **IP address** of the target multiviewer.
6. Click **OK**.

1.4 Editor Screens and Editing a Project using this Manual

A multiviewer project typically contains one or more video wall designs. Each video wall design contains an arrangement of tiles which, in turn, contain various graphical widgets. See the related document 'Orbit - Introduction User Manual' for a full introduction to Orbit and multiviewer applications, including information about projects and managing user access.

'Orbit for Multiviewers' has the tools to define video walls, place graphical widgets and edit their appearance and behavior. This is done in the following Orbit editor screens:

- Wall Editor.
- Tile Editor.
- Theme Editor.

These are covered in the following sections:

- | | |
|-----------|--|
| Section 2 | "Wall Editor" on page 6 |
| Section 3 | "Tile Editor - Tiles and Widgets" on page 73 |
| Section 4 | "Theme Editor - Widget Styles" on page 112 |

Some further aspects of editing projects are covered by:

- | | |
|-----------|--|
| Section 5 | "Widget Behaviours and Bindings" on page 130 |
| Section 6 | "Groups of Widgets" on page 146 |
| Section 7 | "Project Variables" on page 167 |

Some examples are outlined in:

- | | |
|------------|--|
| Section 8 | "Example - Basic Multiviewer Video Wall" on page 188 |
| Section 9 | "Examples - UMDs and Border Tally" on page 228 |
| Section 10 | "Examples - Other On-screen Functionality" on page 285 |

Terminology and lists of Widgets, Behaviours and Bindings are given in:

- | | |
|------------|---|
| Appendix A | "Terminology" on page 318 |
| Appendix B | "List of Widgets" on page 321 |
| Appendix C | "List of Behaviours and Bindings" on page 332 |

2 Wall Editor

Wall Editor

| | |
|--|----|
| Introduction | 7 |
| How to Open and Create Walls | 8 |
| Create a New Video Wall using the Walls Icon | 9 |
| 'Wall file name' and the 'Wall Name' | 11 |
| Create a New Wall via the Project View | 11 |
| Open an Existing Video Wall | 12 |
| Open Existing Video Wall via Project View | 12 |
| Introduction to The Wall Editor Screen | 14 |
| Wall Editor Tool Bar | 15 |
| How to Assign Multiviewer Outputs to a Wall | 16 |
| Assign Outputs Editor Screen | 16 |
| Outputs Banner | 18 |
| Assigning Display Outputs | 19 |
| Display Output Properties | 21 |
| Display Wall Properties | 22 |
| Selecting a Color Value | 24 |
| Display Output Context Menu | 25 |
| Tile Arrangement on a Wall | 28 |
| The Tile Grid and Fine Grid Modes | 28 |
| Wall Editor - Tile Grid Mode | 31 |
| Add Tile Grid Example | 33 |
| Adding a Tile Grid - Using "Output:" drop-down box | 34 |
| Adding a Tile Grid - using Grid Rows and Columns | 35 |
| Adding a Tile Grid and Adjusting Tile Spacing | 37 |
| Resize a Tile in Tile Grid Mode | 37 |
| Add Current Grid, '1+20 Tile' Grid | 40 |
| Reposition a Tile in Tile Grid Mode | 40 |
| Tile Grid Mode - The Clear and Fill Output Controls | 41 |
| Tile Grid Mode - Select, Delete, Resize and Fill | 43 |
| Wall Editor - Fine Grid Mode | 46 |
| Arranging Tiles | 48 |
| Fine Adjustment of Tiles on a Wall | 51 |
| Fine Position and Size Adjustment | 52 |
| Layering (Z order) of Tiles | 53 |
| Fine Adjustment Example: | 54 |
| Add a Custom Tile to Wall | 55 |
| Wall Tile Properties | 57 |
| Assign Inputs to a Wall | 64 |
| Automatic Assignment of Inputs | 64 |
| Manual assignment of video inputs | 67 |
| Wall Layouts | 68 |
| How to Create a new Wall Layout | 68 |
| Switch Wall Layouts in the Orbit Wall Editor | 70 |
| Switch Wall Layouts in Orbit in Run Mode (MV-800-DT License) | 71 |
| Switch Wall Layouts on a Multiviewer Unit | 71 |

2.1 Introduction

This sections describes multiviewer video wall editing

The Orbit application is used to configure, save and load a configuration for a product unit, for example, a multiviewer video wall configuration.

Wall designs and tile arrangements are managed in Orbit's **Wall Editor** screen, which contains a set of tools, graphic editor screens and properties boxes.

Up to 12 wall designs are supported per Orbit project.

Wall designs may be prepared off-line, without being connected to a multiviewer unit. Once prepared, they are pushed to a multiviewer to be deployed. A wall design may also be pulled from a multiviewer, edited and pushed back.

Creating a new or opening an existing wall design accesses the **Wall Editor** screen in Orbit.

Wall design steps:

The preparation of a wall design includes the following steps:

- Define an arrangement of monitor displays that form the video wall.
- Connect multiviewer display outputs to monitor displays.
- Define a tile arrangement for each display output.
- Define the type of tile at each tile position, including pre-built and custom tiles.
- Create different graphical widget styles for use on tiles, as required.
- Connect multiviewer video inputs to each video tile.
- Create and edit any alternate wall layouts.

2.2 How to Open and Create Walls

The Orbit **Wall Editor** screen allows users to define the layout of a video wall. The editor screen is accessed from the Orbit multiviewer project screen and is used to create, review or edit a video wall design.

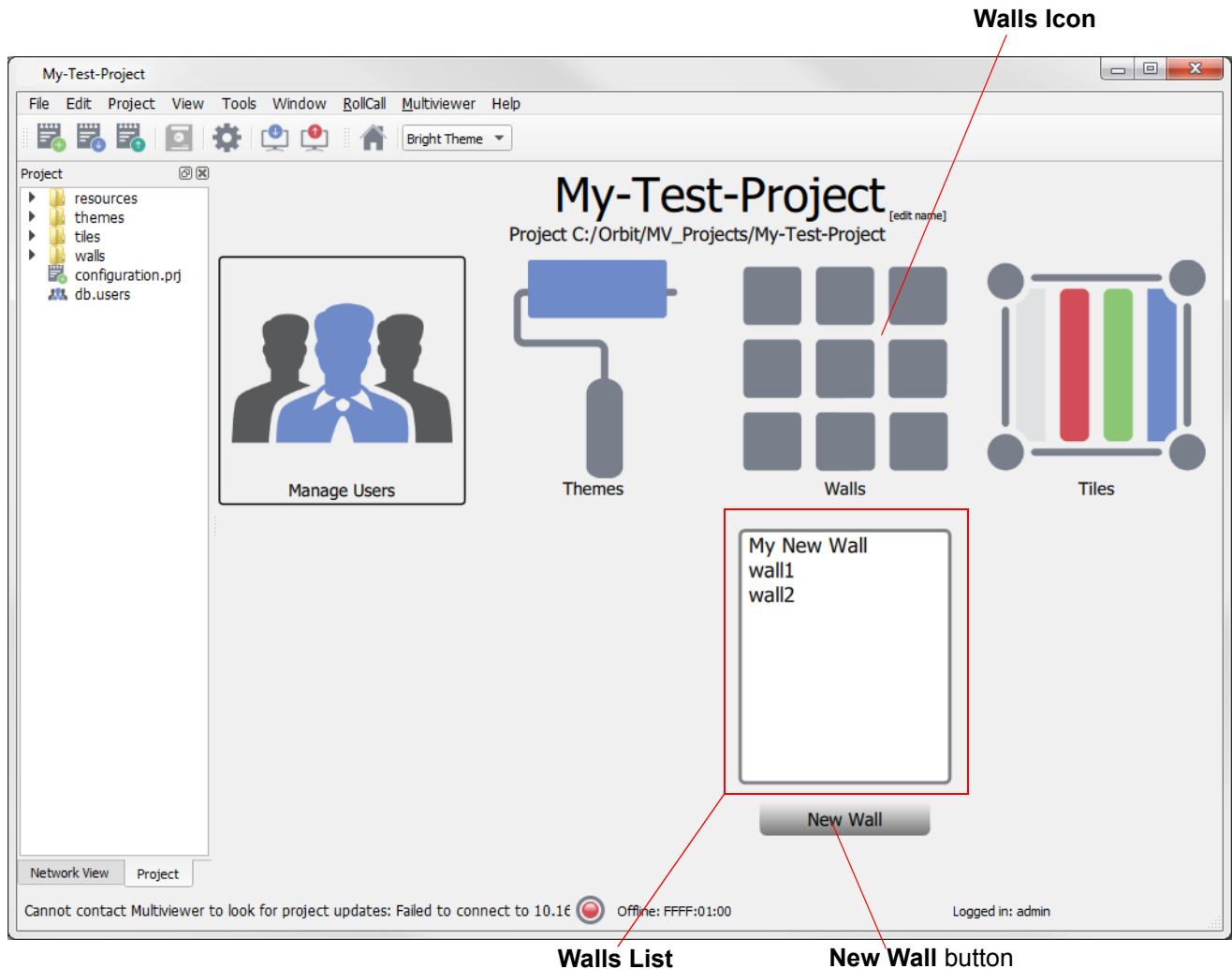


Figure 5 Orbit Project Screen with Walls Icon

Wall designs can be accessed in different ways.

Click the **Walls Icon** in a Orbit **Project Home** screen, see Figure 5.

A list of wall designs appears below the icon. A wall design may be accessed from this list of walls or via the **New Wall** button.

Note: For a new project, the list of walls will be empty.

The **Project View** provides another way to access new or existing wall designs.

2.2.1 Create a New Video Wall using the Walls Icon

To create a new video wall design:

1. In a Orbit **Project Home** screen, click on the **Walls** icon.
2. Click on the **New Wall** button, located below the Walls list.
A **New Wall** dialog appears. See Figure 6.

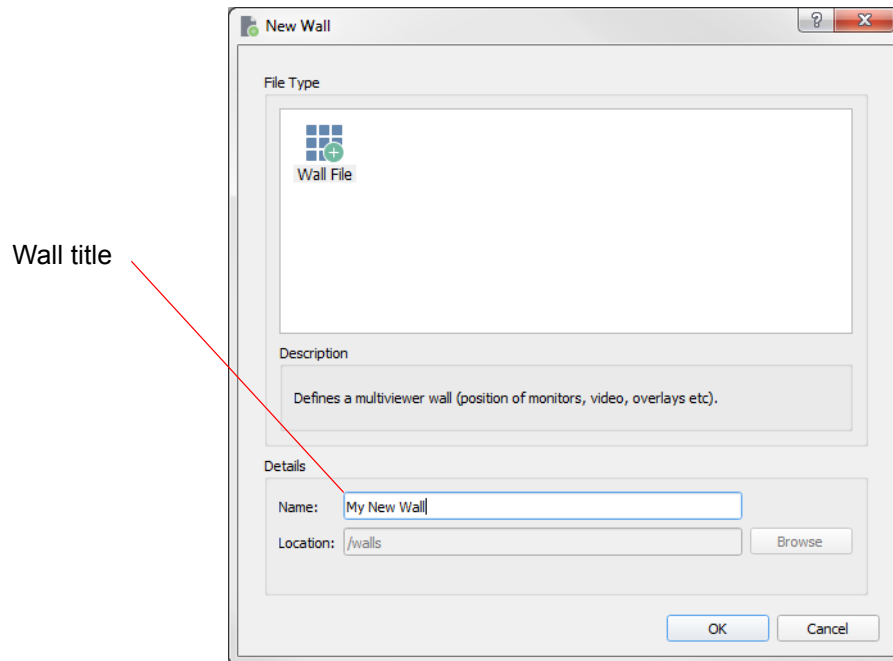


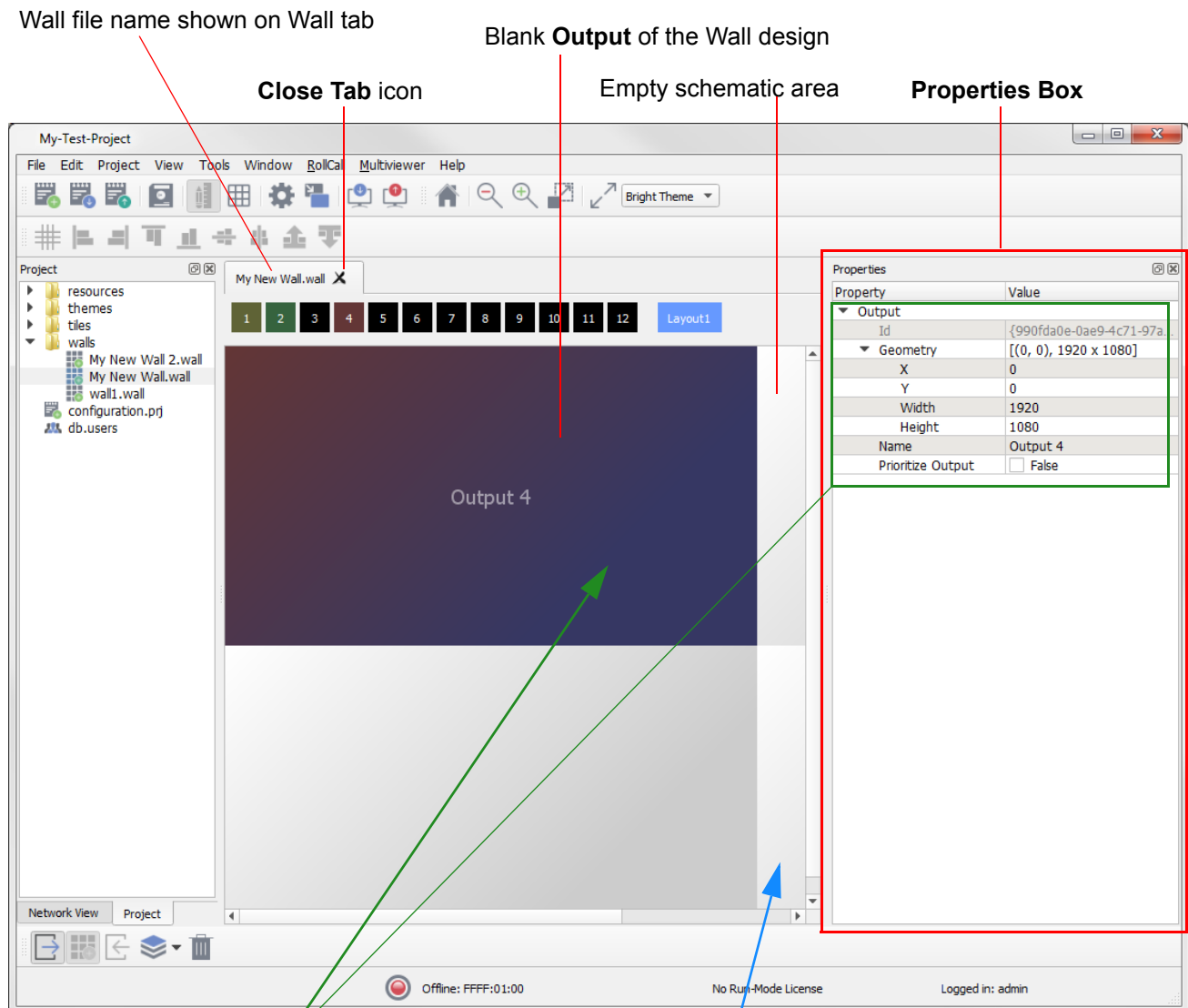
Figure 6 New Wall dialog

3. Enter a short title for the new wall design in the **Name** text box.
Orbit will store all the wall design information in a file of this name with a '.wall' extension. The file name will then appear in the **Project View**, under the '**walls**' item.
4. The **Location** text box shows the location of the wall design file within in the Orbit project, in the '**walls**' sub-folder.
Do *not* change this location.
5. Click **OK** (or click **Cancel** to return to the Orbit project screen).

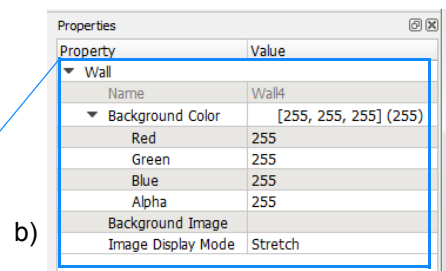
A new wall screen opens and the **Wall Editor** screen tab is displayed, see Figure 7.

Note: All information relating to a video wall design is stored within the Orbit project in a file with a '.wall' suffix.

Note: Do not change the 'Location' of the wall design files.



- a) Click on the blank **Output** of the wall design to see multiviewer **Output Properties** in the **Properties Box**



- Click on the empty, gray schematic area below the wall design to see **Wall Properties** in the **Properties Box**

Figure 7 Wall Editor Tab - with Output and Wall Properties

A new, blank **Wall** design has been created.

Wall properties are shown on the right-hand side.

- Click in the **Output** area to see the **Output Properties** (i.e. the properties of the multiviewer output comprising the wall). See Figure 7a.
- Click in the gray schematic area below the wall design to see the **Wall Properties** (i.e. the properties of the whole wall). See Figure 7b.
- Save changes by clicking the **Save File** icon in Orbit screen's main tool bar.
- Click **File > Save Project** to save the whole Orbit project.

A **Wall Editor** tab may be closed by clicking on the **Close Tab** icon. If changes have been made in the tab, a prompt is shown, asking to save any changes made.

A new, blank wall design has now been created.

Note:

A Orbit project may contain more than one wall design.

A maximum of 12 walls are supported.

2.2.2 'Wall file name' and the 'Wall Name'

Note:

'Wall file name' and the 'Wall Name':

Think of the **'Wall file name'** as a short descriptive title.

Think of the **'Wall Name'** as an internal identifier.

'Wall file name':

- A **Wall** design is stored in a **Wall** file in a Orbit project.
- A **Wall** file:
 - Is shown in the **Project View** under the **'walls'** item.
 - Has a **'wall'** extension.
 - May be imported into other Orbit projects.
- The **'Wall file name'**:
 - Is editable by the user.
 - Must be unique within a Orbit project.
 - Appears on the **Wall Editor** tab.

'Wall Name':

- Each **Wall** has a **Name** property.
- The **Name** property is:
 - Set to **"Wall1"**, **"Wall2"**, etc. automatically by Orbit.
 - Unique within a Orbit project.
 - Used to refer to a **Wall** design when deployed on a multiviewer device. For example, when changing **Wall** layout with a RollCall command.

2.2.3 Create a New Wall via the Project View

The **Wall Editor** tab screen is labeled with the wall design name and a **".wall"** suffix.

A file with this name appears in the Orbit project, in sub-folder **'walls'**.

The wall file name is shown in the **Project View**.

The **Project View** is a useful way of opening various project design information files, including wall designs.

Note:

Caution:

Do not edit nor delete Orbit project design information files outside the Orbit application.

In the **Project View**:

1. Right-click on the '**walls**' sub-folder item in the **Project View**.
A selection menu appears. See Figure 8.

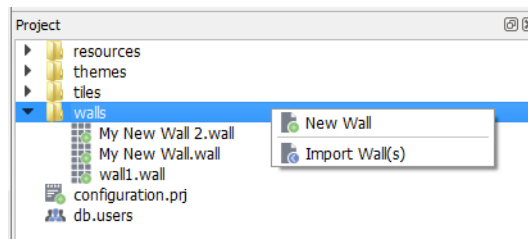


Figure 8 New Wall via the Project View

2. Click **New Wall**.
The **New Wall** dialog appears, see Figure 6 on page 9.
3. Proceed as described in Section 2.2.1, "Create a New Video Wall using the Walls Icon" on page 9
to create a new, blank wall design.

2.2.4 Open an Existing Video Wall

1. In an Orbit **Project Home** screen, click the **Walls** icon.
A list of **Wall** designs appears below the icon.
2. In the **Walls List**, click on the wall design name to be opened.
The **Wall Editor** screen tab is displayed.

The existing **Wall** design has been opened.

2.2.5 Open Existing Video Wall via Project View

The **Wall Editor** tab screen is labeled with the wall design name and a ".wall" suffix.
A file with this name appears in the Orbit project, in sub-folder '**walls**'.
The wall file name is shown in the **Project View**.

The **Project View** is a useful way of opening various project design information files, including wall designs.

Note:**Caution:**

Do not edit nor delete Orbit project design information files outside the Orbit application.

1. In an Orbit **Project Home** screen, expand the **Project View** items to see the contents of the '**walls**' sub-folder item. See Figure 9.

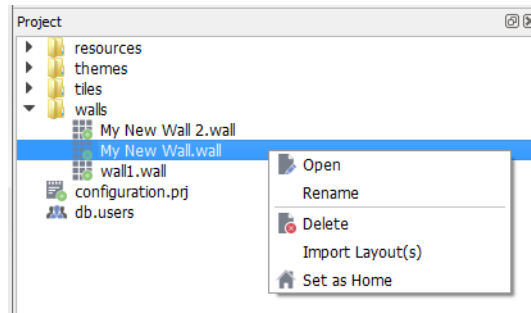


Figure 9 New Wall via the Project View

2. Locate the design file for an existing wall design (the file name ends in '.wall'). And select the '.wall' file name in the **Project View**
3. Then either:
 - a Right-click on the file name to reveal a menu; then click **Open** in the menu. See Figure 9. The wall design is opened in the **Wall Editor**.Or:
 - b Double-click on the file name. The wall design is opened in the **Wall Editor**.

Note:**Caution:**

Only use the **Project View** when you are familiar with the Orbit tool because it is possible to rename or delete a project file via the **Project View**.

2.3 Introduction to The Wall Editor Screen

The **Wall Editor** screen contains a main wall design area, a properties box, several new main tool bar icons and a new **Wall Editor** tool bar, located at the lower left. Figure 10 highlights some of the features of the **Wall Editor** screen.

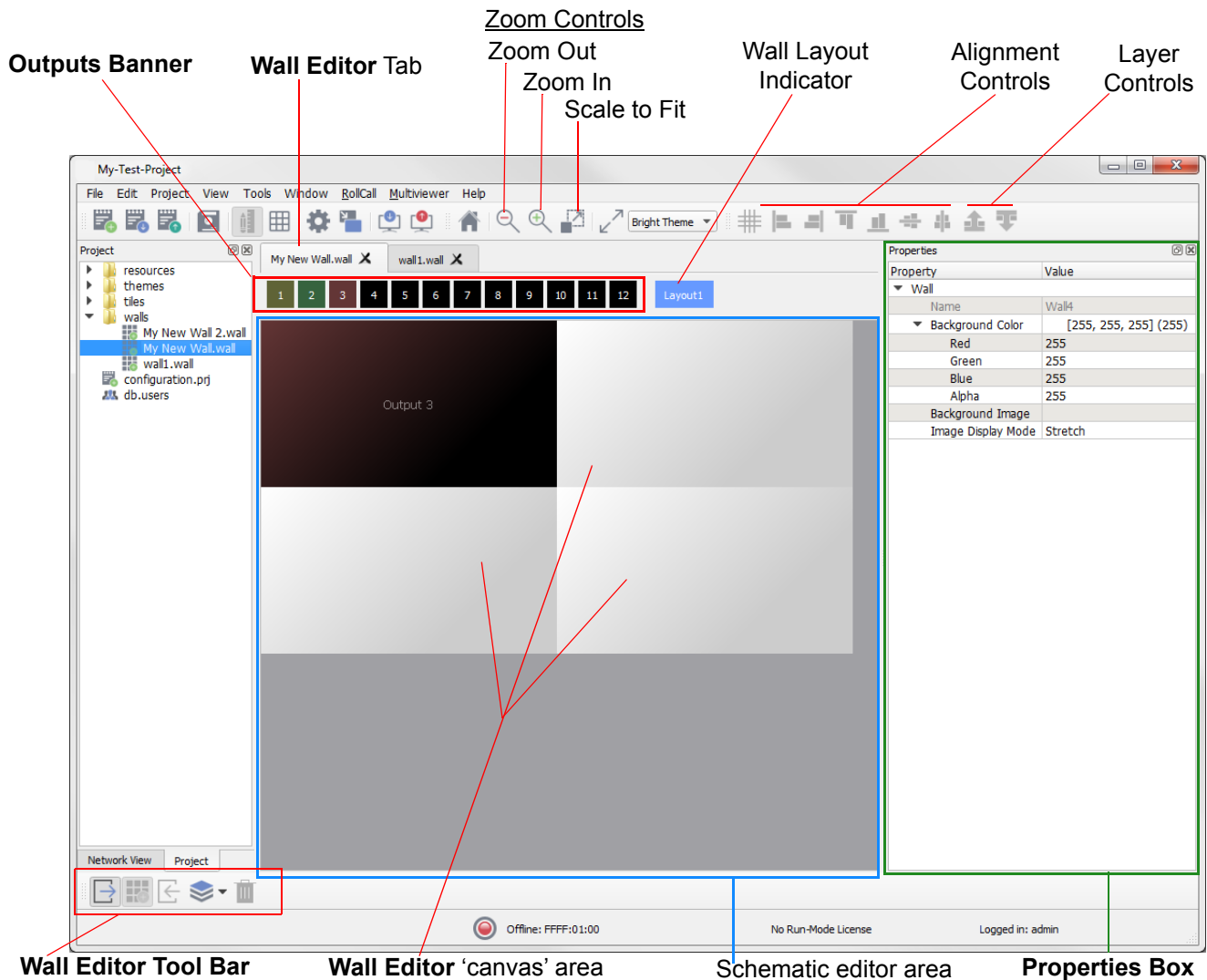


Figure 10 Wall Editor screen and a New Wall Design

Main menu and tool bar items are described in the 'Orbit - Introduction User Manual'.

An **Outputs Banner** is located beneath the **Wall Editor** tab. This shows which multiviewer outputs are available and which are used on wall designs in the project. Multiviewer outputs can be selected by clicking on icons in the banner. More than one output may comprise a wall design.

The schematic editor area is a canvas upon which the wall design is laid out. It contains the 'canvas' area where the wall is designed.

The **Wall Editor** screen is where graphical wall design schematics are edited and prepared.


The **Wall Editor Tool Bar** contains tools for wall layout design.

The precise **Wall Editor** screen appearance depends upon whether the wall is newly-created, or whether it exists and has already been designed.

The general screen appearance depends on which **Wall Editor Tool Bar** icon is selected.

The **Properties Box** shows the properties of a selected wall item. Properties can be viewed and edited in the properties box. Select a properties item with a single click.

The relative sizes of the schematic editor area, **Properties Box** etc can be changed by:

1. Positioning the cursor between the two screen areas.
2. When the cursor changes shape to , dragging the cursor to resize the areas.

The **Wall Editor** screen also offers **Zoom Controls**, which enable the user to display the wall's area of interest while editing.

A **Wall Layout** indicator shows which wall layout is being edited.

2.3.1 Wall Editor Tool Bar

The **Wall Editor** tool bar provides a set of tools to edit the wall design.

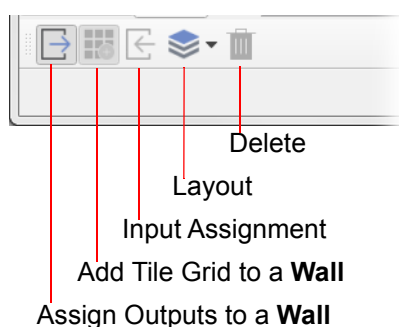


Figure 11 Wall Editor Tool Bar Icons

| Tool Bar Icon | Description |
|---------------------------------|--|
| Assign Outputs to a Wall | <p>Allows a user to define how and how many monitor displays form a video wall design. The monitor display arrangement is defined.</p> <p>See Section 2.4 “How to Assign Multiviewer Outputs to a Wall” on page 16.</p> |
| Add Tile Grid to a Wall | <p>Allows a user to define tile positions on each monitor display. This is done with editable tile grids, which define regular arrays of tile. This forms a coarse grid of tile positions.</p> <p>Tile positions can be finely adjusted on a wall, as required, using a fine grid in Fine Grid mode.</p> <p>Wall editing is carried out in the Wall Editor and may be done on a coarse tile-grid or on a fine grid:</p> <ul style="list-style-type: none"> • See Section 2.6 “Wall Editor - Tile Grid Mode” on page 31. • See Section 2.7 “Wall Editor - Fine Grid Mode” on page 46 and see Section 2.8 “Fine Adjustment of Tiles on a Wall” on page 51. |
| Input Assignment | <p>Assign multiviewer inputs to wall tiles.</p> <p>See Section 2.9 “Add a Custom Tile to Wall” on page 55.</p> |
| Layout | <p>More than one wall layout can be designed and viewed in the editor.</p> <p>See Section 2.12 “Wall Layouts” on page 68.</p> |
| Delete | <p>Deletes a selected graphical item in the wall schematic editor.</p> |

Table 2 Wall Editor Tool Bar Functions

2.4 How to Assign Multiviewer Outputs to a Wall

The first step in a new wall design is to define how many monitor displays the new wall consists of and their arrangement.

The monitor arrangement of an existing wall design may also be edited.

A multiviewer's display outputs need to be assigned to each monitor display on the video wall.

2.4.1 Assign Outputs Editor Screen

This Orbit editor screen allows multiviewer display outputs to be connected to a wall design, see Figure 12. The wall is best viewed zoomed-out when assigning outputs.

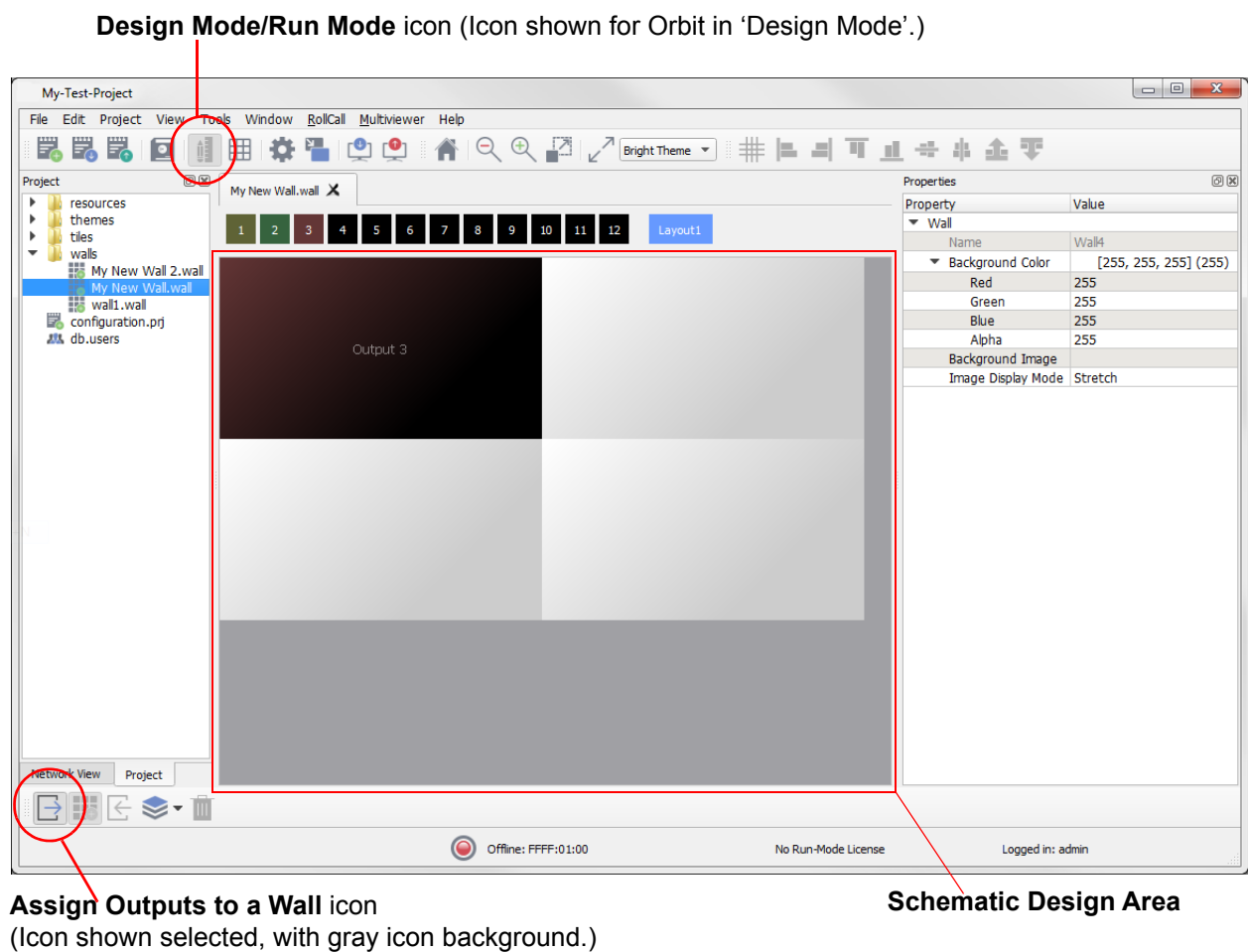


Figure 12 “Assign Display Outputs to Wall” editor screen (Zoomed out)

If your Orbit screen does not resemble Figure 12, ensure that:

1. The Orbit application is in 'Design Mode' (rather than 'Run Mode'); Click on the **Design/Run Mode** icon in the main tool bar to toggle between modes. The icon must have a gray background, as shown in Figure 12, for 'Design Mode'.
Note: When the editor is in '**Design Mode**' the icon shows a 'Run Mode' message as the cursor hovers over the icon.
2. The **Assign Outputs to a Wall** icon in the **Wall Editor** tool bar is selected (i.e. it has a gray icon background). Click the icon to select/deselect it.
3. Zoom out with the **Zoom Out** (-) control (main tool bar) and use the **Scale-to-fit** control to best see a wall in the design area.

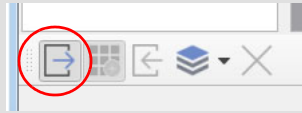
Figure 12 shows the “Assign Display Outputs to Wall” mode of the **Wall Editor**.

The first available multiviewer display output is assigned to a new wall (i.e. This is Output 1, for a new wall in a *brand new* Orbit multiviewer project).

Note:

Wall Editor Tool Bar and assigning outputs:

When viewing the “Assign Display Outputs to Wall” mode of the **Wall Editor**, the **Assign Outputs to Wall** icon should be selected. If this is not the case, then re-click on the **Assign Outputs to Wall** icon.



Assign Display Outputs to Wall icon when assigning outputs to a wall.

Note:

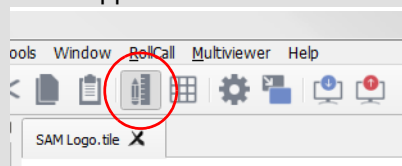
Design Mode / Run Mode icon appearance:

Wall editing is done in ‘Design Mode’.

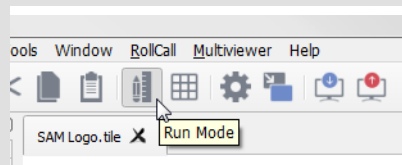
The appearance of the icon is shown below for Orbit in ‘Design Mode’ and for Orbit in ‘Run Mode’:

- When Orbit is in ‘Design Mode’:

Icon appearance:

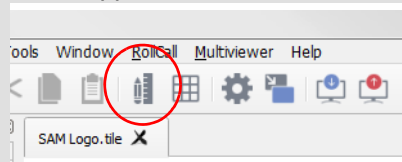


Cursor over icon:

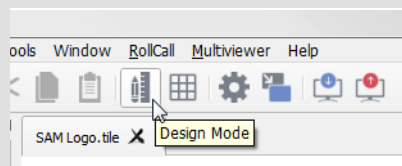


- When Orbit is in ‘Run Mode’:

Icon appearance:



Cursor over icon:



2.4.2 Outputs Banner

The **Outputs Banner** shows information about multiviewer outputs and assignment of an output to a wall. It shows which multiviewer outputs are available and which are already used on wall designs. See Figure 13.

Output 1 - shown as being used on another **Wall**.

Output 2 - shown as being used on yet another **Wall**.

Outputs 3, 4 and 5 - used on the **Wall** shown.

Outputs 6 to 12 - available but currently unused in a **Wall** design.

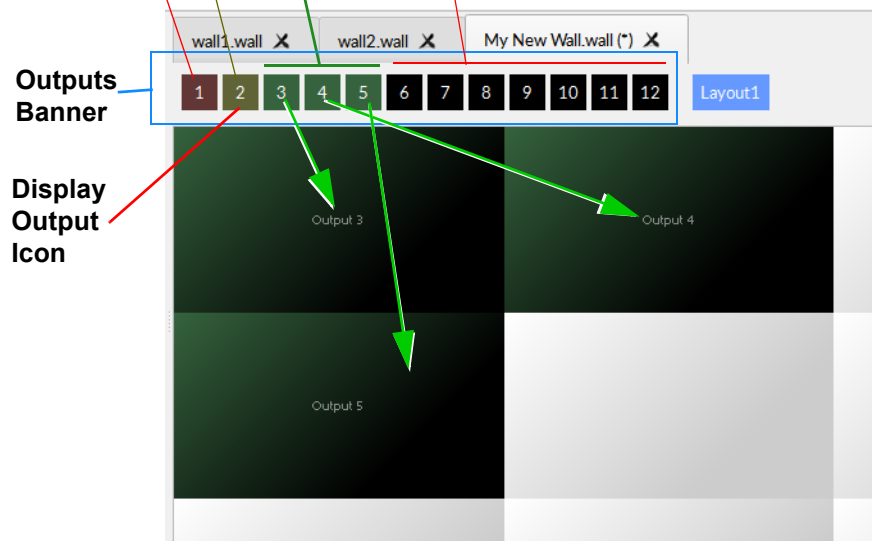


Figure 13 Outputs Banner and Display Output Icons

Note:

Display Output Icons:

- **Black** if the multiviewer output is available, licensed and is not already assigned to a video wall design in the Orbit project.
- **Colored** when the output is assigned to a wall, see Figure 14.
This indicates that the output is already assigned to a wall design in the project and it is unavailable for further assigning.
- All 12 display outputs of the multiviewer are available in a Orbit project.

Note:

Display Outputs and Licenses:

All 12 display outputs may be assigned in a Orbit project.
Only licensed display outputs, however, are shown by any multiviewer unit.

Assignment of multiviewer display outputs is described in Section 2.4.3 “Assigning Display Outputs” on page 19.

2.4.3 Assigning Display Outputs

The **Wall Editor**'s schematic design area represents the entire video wall being designed. By default, for a new wall design, one display device using one multiviewer output is assumed. More multiviewer outputs may be added (assigned), as described in this section.

A video wall design may be formed from one or more monitor display devices placed adjacent to each other. A multiviewer may typically have four or more **Display Outputs** and a new multiviewer project reflects this by showing four or more available outputs, depending on the multiviewer model and licensing available.

The current maximum number of outputs supported by Orbit is 12.

The current maximum number of walls is 12.

Note: By default a MV-800, a MV-820 and a MV-850 multiviewer have 4-off outputs licensed. More outputs may be purchased and licensed from Grass Valley.

To assign multiviewer four (2x2) outputs to a wall:

1. Drag and release a **Display Output Icon** from the **Outputs Banner** onto the schematic design area. See Figure 14a.
2. Drag each remaining output icon onto the wall. See Figure 14b, c and d.
For example, a resulting 2x2 arrangement of outputs is shown in Figure 14d.

Form the display devices arrangement for your video wall. Figure 14 shows a 2x2 example.

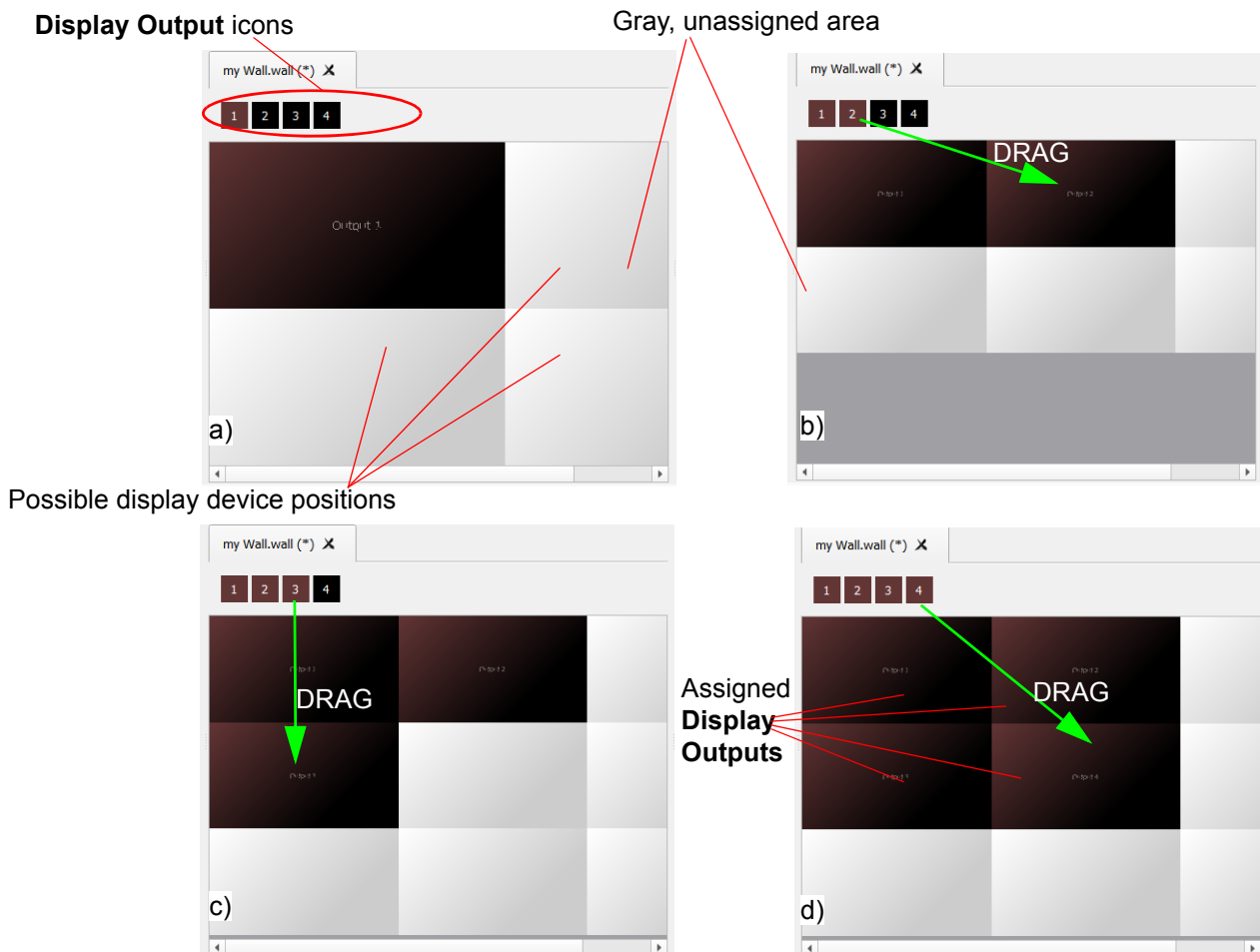


Figure 14 Wall Schematic Editor Area - Assigning Display Outputs:

- a) Output 1 assigned on the editor screen.
b) Output 2 dragged on and Assigned.
c) Output 3. d) Output 4.

Assigning Display Outputs:

To edit an arrangement of assigned multiviewer **Display Outputs**:

3. Click on an assigned **Display Output**, to select it.
4. Drag it to a new, available **Display Output** position and release.

To UNassign a **Display Output**:

5. Select a **Display Output**.
6. Then either:
 - Click the **Delete** icon in the **Wall Editor Tool Bar**.
 - or use the keyboard 'delete' key.
 - or drag the **Display Output** to the **Outputs Banner** and release it.

To re-assign a **Display Output**:

7. Drag the **Display Output Icon** onto the wall to an available output position.

When all **Display Output**: assignments have been made:

8. Click the **Save File** icon to save the wall design changes.

Viewing Output Properties or Wall Properties:

To view the properties of a **Display Output** in the **Properties Box**:

9. Click on a **Display Output** to select it.
The properties of the output are shown.
See Section 2.4.4 "Display Output Properties" on page 21.

To view the properties of the **Wall** in the **Properties Box**:

10. Click in the gray unassigned area.
The properties of the wall are shown.
See Section 2.4.5 "Display Wall Properties" on page 22.

To exit the **Assign Display Outputs** screen:

When all output assignments to the wall are complete, leave the "Assign Display Outputs to Wall" mode of the **Wall Editor** by doing the following:

11. Click the **Assign Outputs to Wall** icon to deselect it.
(When the icon is deselected, the icon background changes, see Figure 15.)



Icon deselected



Icon selected

For "Assign Display Outputs to Wall"
mode of the **Wall Editor**

Figure 15 Deselected "Assign Outputs to Wall" icon

After display outputs are all assigned to a wall, the next step is to layout tiles on the wall, see following sections.

2.4.4 Display Output Properties

In the **Wall Editor** screen, click on a **Display Output**. This selects the display output and the **Property Box** shows the display output properties. See Figure 16.

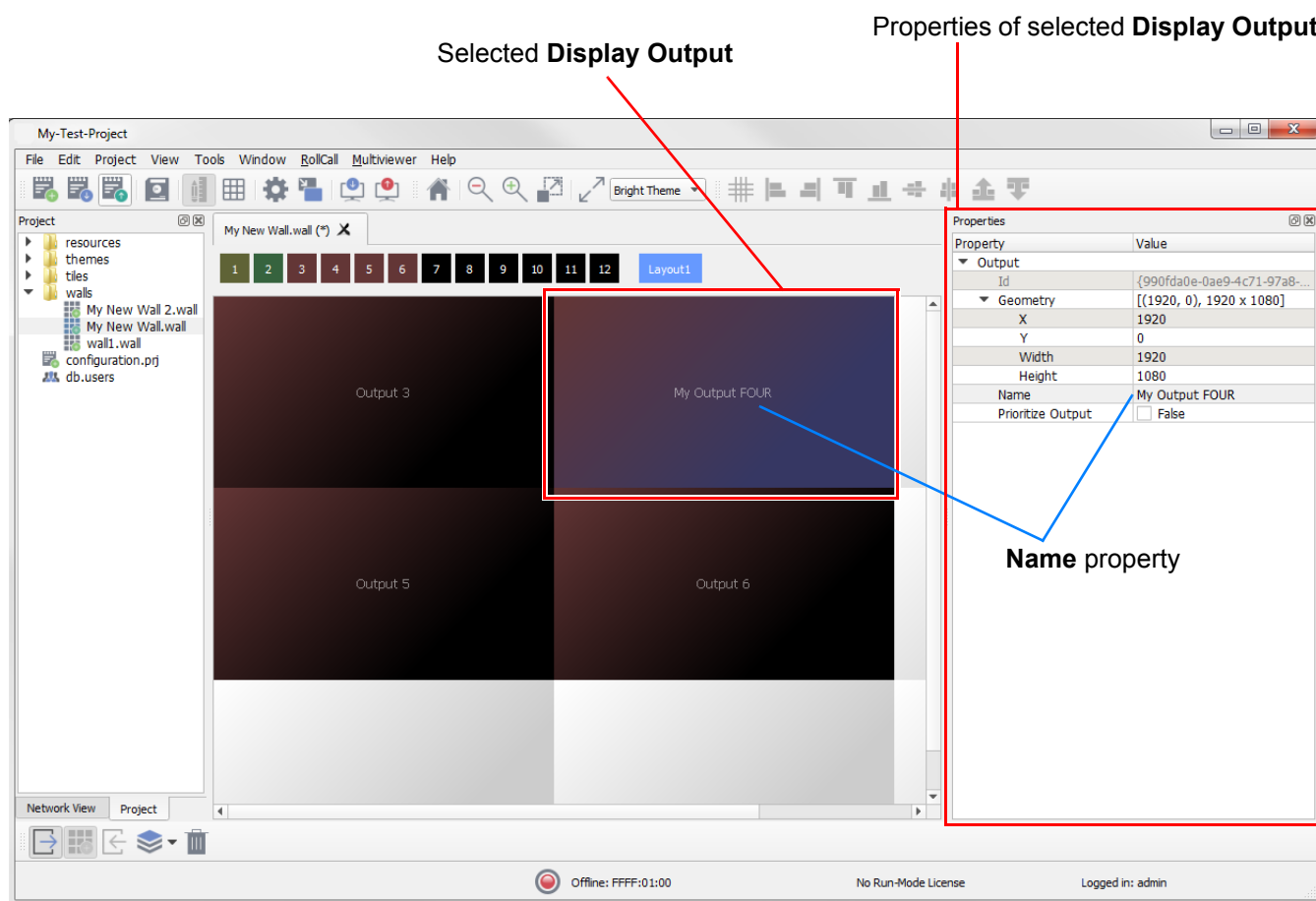




Figure 16 Display Output Properties

Expand or collapse items in the **Properties Box** using the in-line triangular expand/collapse icons ( ).

| Property | Description |
|-----------|---|
| Id | A GUID name generated by Orbit. Not editable. |
| Geometry: | Geometry properties are summarized in brackets: [(X, Y), Width x Height] |
| X | Video wall horizontal position of the top left-hand corner of the <i>display output</i> . Measured in lines. Note: The top left-hand corner of the <i>wall</i> is pixel position 0. |
| Y | Video wall vertical position of the top left-hand corner of the <i>display output</i> . Measured in pixels. Note: The top left-hand corner pixel of the <i>wall</i> is on line 0. |
| Width | Number of pixels in a line for the display output. |
| Height | Number of lines for the display output. |

Table 3 Display Output Properties

| Property | Description |
|---------------------------|---|
| Name | Text box. Enter a text label for the Display Output . The Name text label appears in the Wall Editor Area . |
| Prioritize Overlay | Check box. Select to prioritize rendering of graphical items for a particular output. Note: The updating of rendered graphical items on the video wall (for example, audio bars and clocks) may slow down if many video streams are displayed. The Prioritize Overlay property is used to give rendering priority and increase the update rate for a particular output screen. |

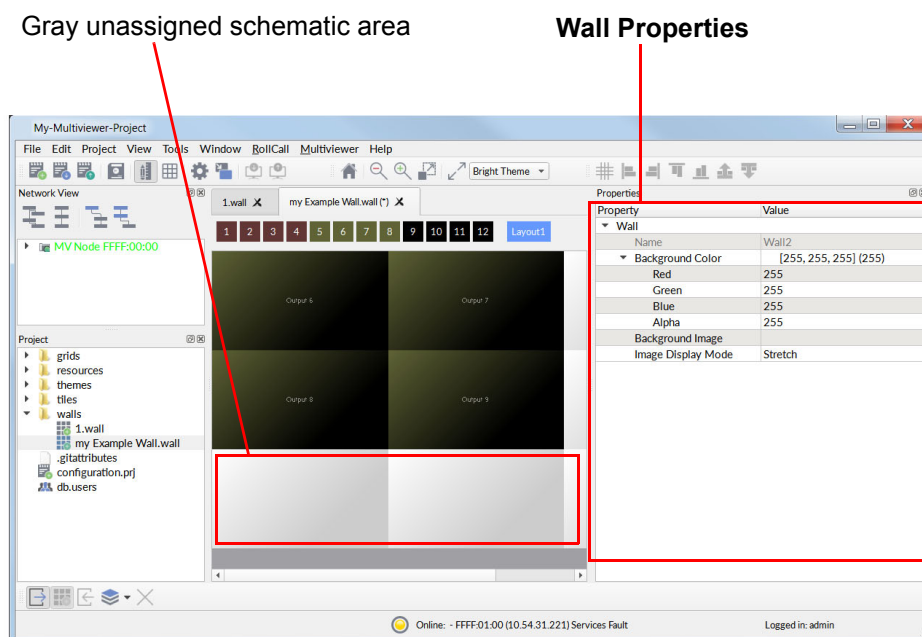
Table 3 Display Output Properties (Continued)

After making changes, save them by clicking the **Save File** icon in the main tool bar.

Note: All output displays must be the same size and resolution.

2.4.5 Display Wall Properties

Clicking in the surrounding gray, unassigned, area of the **Wall Editor** deselects all **Display Outputs** and the **Properties Box** displays **Wall Properties**.

**Figure 17 Display Wall Properties**

| Property | Description | | | | | | | | |
|--------------------|--|-----|---|-------|---|------|--|-------|--|
| Name | <p>Wall name, assigned by Orbit. Not editable.</p> <p>The Wall Name and the Layout Name are used when selecting a wall layout remotely.</p> | | | | | | | | |
| Background Color | <p>Background wall color.</p> <p>The color value is shown in the following format:</p> <p>[Color sample] (Red, Green, Blue) (Alpha)</p> <p>See Section 2.4.6 “Selecting a Color Value” on page 24.</p> <tr> <td>Red</td><td>Value for Red component. (0-255)</td></tr> <tr> <td>Green</td><td>Value for Green component. (0-255)</td></tr> <tr> <td>Blue</td><td>Value for Blue component. (0-255)</td></tr> <tr> <td>Alpha</td><td> <p>Value for the Alpha component. (Opacity, 0-255, of the color: 0 - Fully transparent. 0% 255 - Fully opaque color. 100%)</p> <p>Note: The color property values are expressed as (Red, Green, Blue) (Alpha) number sets. Check the Alpha value is non-zero if your color is not appearing on your widget/tile or wall.</p> </td></tr> | Red | Value for Red component. (0-255) | Green | Value for Green component. (0-255) | Blue | Value for Blue component. (0-255) | Alpha | <p>Value for the Alpha component. (Opacity, 0-255, of the color: 0 - Fully transparent. 0% 255 - Fully opaque color. 100%)</p> <p>Note: The color property values are expressed as (Red, Green, Blue) (Alpha) number sets. Check the Alpha value is non-zero if your color is not appearing on your widget/tile or wall.</p> |
| Red | Value for Red component. (0-255) | | | | | | | | |
| Green | Value for Green component. (0-255) | | | | | | | | |
| Blue | Value for Blue component. (0-255) | | | | | | | | |
| Alpha | <p>Value for the Alpha component. (Opacity, 0-255, of the color: 0 - Fully transparent. 0% 255 - Fully opaque color. 100%)</p> <p>Note: The color property values are expressed as (Red, Green, Blue) (Alpha) number sets. Check the Alpha value is non-zero if your color is not appearing on your widget/tile or wall.</p> | | | | | | | | |
| Background Image | <p>File name box: Specifies the image file in the Orbit Project to use as the video wall background image.</p> <p>The path to the file name is shown in the property value box. The Orbit project folder may be browsed to locate the required image.</p> <p>The image must have been previously imported into the project before it is available as a project resource to use. Import image file(s) into the project via the Project View.</p> <p>To import an image file into a project:</p> <ol style="list-style-type: none"> 1. In Project View, expand the ‘resources’ folder. 2. Right-click on the ‘Images’ folder and click Import File(s). A file explorer dialog opens. 3. Browse to the image file and click Open. <p>The image file appears in the Orbit project’s ‘images’ sub-folder, in the Project View.</p> | | | | | | | | |
| Image Display Mode | <p>Drop-down box. Sets how the wall display area is covered by the image:</p> <ul style="list-style-type: none"> • Stretch - Image is stretched to cover the wall. • Tile - Image is repeated (tiled) to cover the wall. | | | | | | | | |

Table 4 Display Wall Properties

After making property changes, save them by clicking the **Save File** icon in the main tool bar.

2.4.6 Selecting a Color Value

Note:


Showing color values:

Color values may be shown in the **Properties Box** in either of the following ways:

- R,G,B and Alpha decimal color format:

Example 1: [243, 196, 7] (128)

Red= 243, Green= 196, Blue= 7. Alpha= 128.


(I.e.  [243, 196, 7] (128) , a semi-transparent yellow color.)

or

- HTML color format (with hexadecimal numbers, i.e. using digits 0 to F)

Example 2: #ff00ee (6 hexadecimal digits, R G B)

Red= hex FF, Green= hex 00, Blue= hex EE. (And Alpha= hex FF.)

(I.e.  #ff00ff , a solid magenta color.)

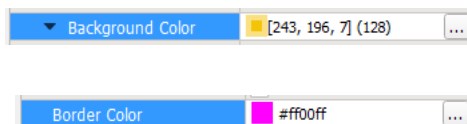
Example 3: #80ff00ee (8 hexadecimal digits, Alpha R G B)

Alpha= hex 80, Red= hex FF, Green= hex 00, Blue= hex EE.

(I.e.  #80ff00ee , a semi-transparent magenta color.)

To select a color for a color property value, either:

1. Click the color property value to select it.



2. Click on the ... **Color Selection Button** (), see Figure 18. Select the color with the **Select Color Dialog**, see Figure 19. Click **OK**.

or

3. Enter **Red**, **Green**, **Blue** and Alpha property values explicitly.

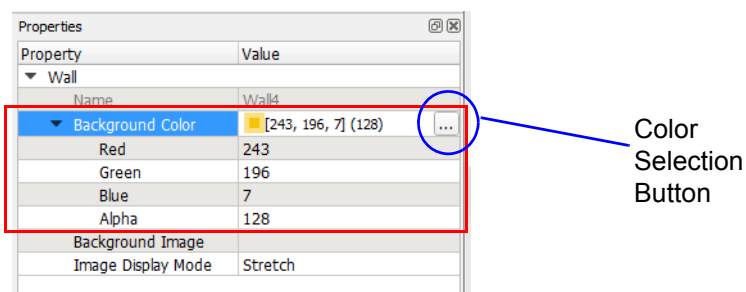


Figure 18 (Wall Background) Color Property

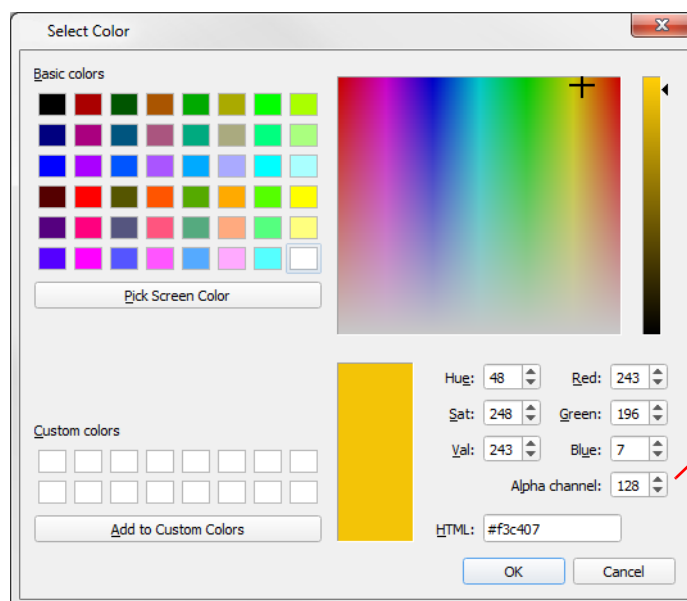


Figure 19 Select Color Dialog Box

Example:

- 50% transparent background color property example:



2.4.7 Display Output Context Menu

When in the **Wall Editor**, a context-sensitive menu may be obtained by right-clicking in the schematic area. The menu provides quick access to some menu bar or tool bar functions:

1. Click on a **Display Output** in the **Wall Editor** to select the output.
2. Right-click on the output.
A pop-up menu appears, see Figure 20 and Table 5.

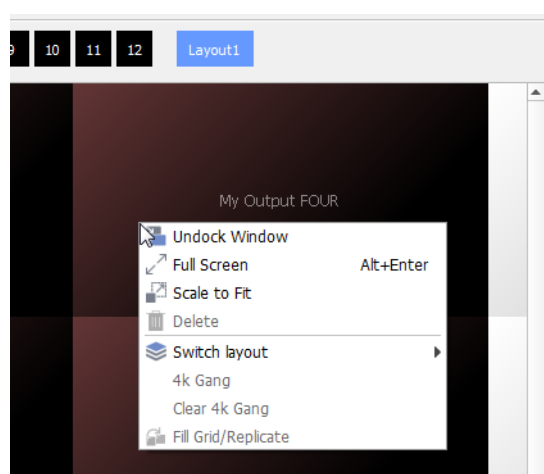
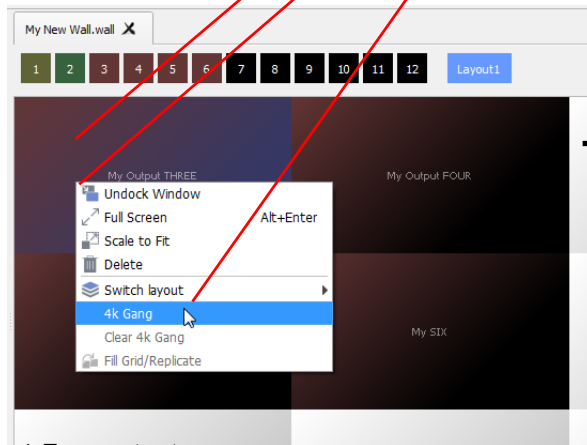


Figure 20 Right-Click Display Output Pop-up Menu

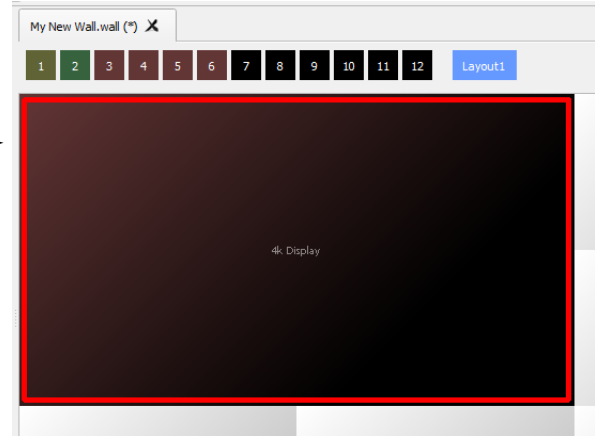
| Menu Item | Description |
|----------------------------|---|
| Undock Window | Undocks / re-docks a Orbit graphical editor tab from the Orbit window and displays it in a separate window. For example, this allows the window to be displayed on a second PC monitor. |
| Full Screen | Option available when a Wall or Tile Editor is open. Click to view the current graphical object full-screen. To return from full-screen mode, press [Alt]+[Return] , i.e. the “Alt” key and “Return” key together. |
| Scale to Fit | Resize (scale) a graphical schematic to fit the Orbit graphical schematic editor screen area. |
| Delete | Deletes the output assignment on the wall. |
| Switch layout | Allows user to select another wall Layout. There can be more than one wall layout for each output. |
| 4k Gang | Gangs together four outputs onto one wall design canvas. See Figure 21a,b. |
| Clear 4k Gang | Removes a 4K Gang. See Figure 21c,d. |
| Fill Grid/Replicate | Copies the selected tile to all wall tile grid positions. |

Table 5 Output Context Menu Items

1. Click in the *top-left Display Output*, to select it.
2. Right-click to see context menu.
3. Click **4K Gang**.
4. Click **Save**.

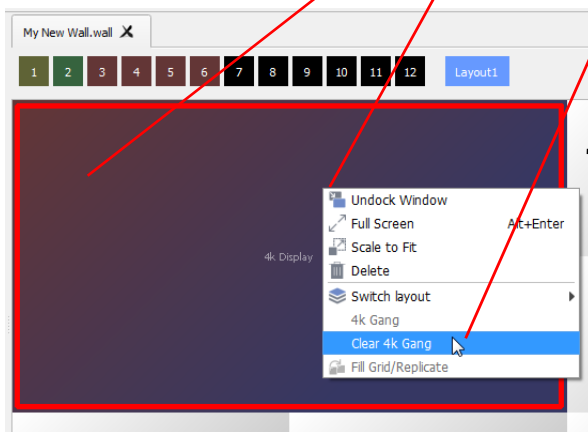


a) Four outputs.

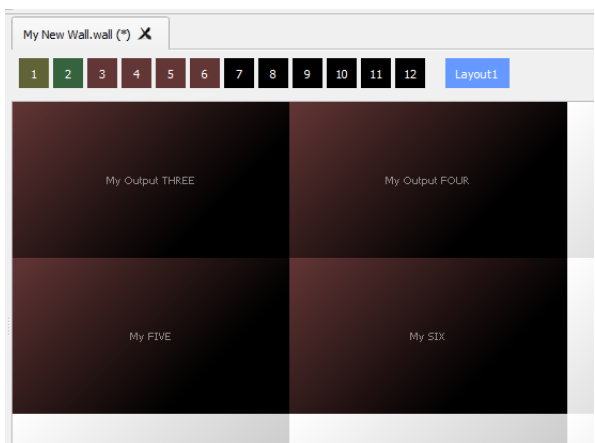


b) Ganged outputs for 4K.

5. Click in the **4K** output, to select it.
6. Right-click to see context menu.
7. Click **Clear 4k Gang**.



c) 4K output.



d) Four outputs.

Figure 21 Display Output Ganging

a) Four outputs shown as four quadrants

b) 4k Gang: Four Outputs are ganged together, in one design canvas.

c) 4K output.

d) Four quadrants.

2.5 Tile Arrangement on a Wall

When multiviewer display outputs have been fully-assigned to the multiviewer video wall (see Section 2.4 “How to Assign Multiviewer Outputs to a Wall” on page 16), the next step is to give the wall an arrangement of tiles.

For a new wall design, there will be no tiles added to a wall design. The first step is to determine the required tile arrangement on each output.

Pre-defined **Tile Grid** templates may be added to each display output area of a wall. Orbit supplies various **Tile Grid** templates and others can be created and saved. A **Fine Grid** editor mode then allows any fine adjustment of tile sizes and position that might ultimately be required.

Thus, tile **positions** may be added and edited:

- En bloc, in coarse grids, via a **Tile Grid** edit mode.
- Individually and finely via a **Fine Grid** edit mode.

Once added, the basic **Tile Grids** can be modified and edited (see examples in Figure 22), and saved for further use.

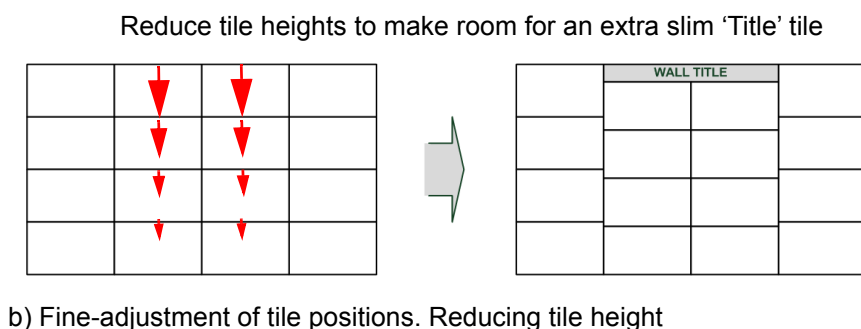
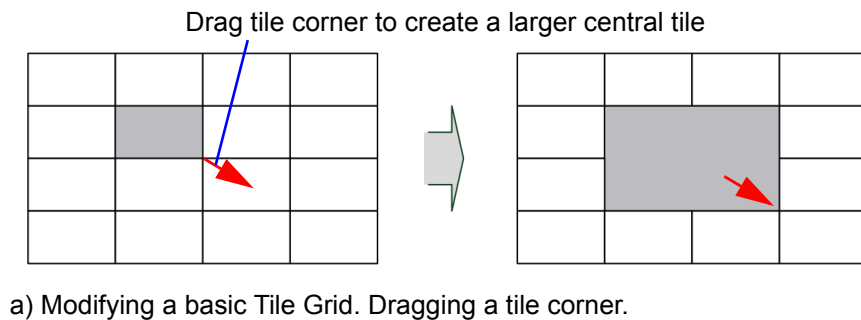


Figure 22 “Examples of editing tile positions:

a) Modifying a basic Tile Grid.

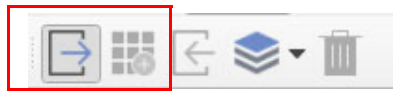
b) Fine-adjustment of tile position.

2.5.1 The Tile Grid and Fine Grid Modes

The Orbit **Wall Editor** has two edit modes for managing tiles on a wall:

- Tile Grid** edit mode - for adding tile arrangements and quick positioning of tiles.
- Fine Grid** edit mode - for fine adjustment of tile position and size, and adding any individual, ‘off-grid’ tiles.

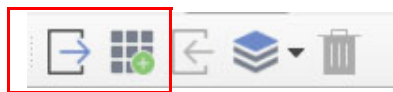
The **Wall Editor Tool Bar** determines which edit mode is shown, see Figure 23.



Assign Outputs



Tile Grid edit mode



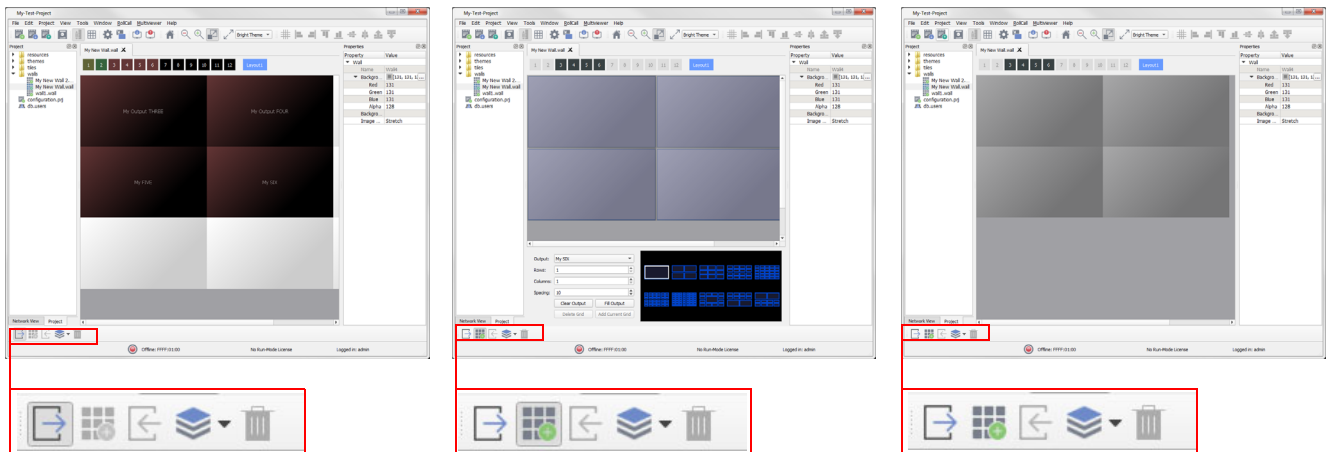
Fine Grid edit mode

Figure 23 Wall Editor Tool Bar and Wall Editor Modes

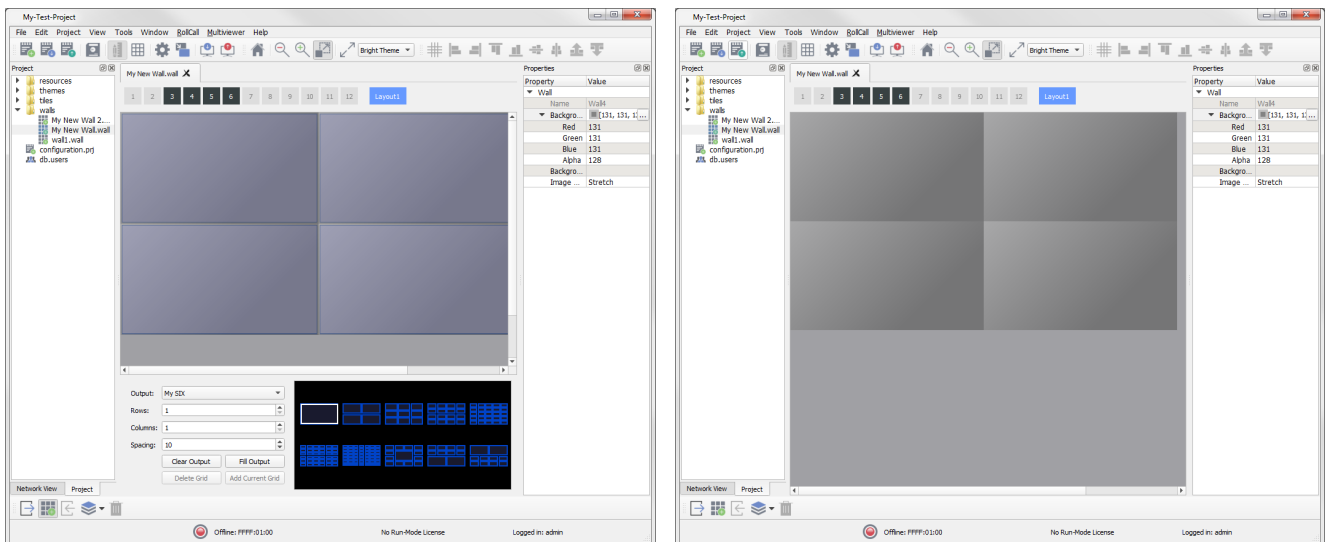
To see these two modes in the **Wall Editor**:

1. Open an output-assigned wall design in the **Wall Editor** in Orbit.
Use the zoom controls (main tool bar) to see the entire video wall in the editor screen.
2. Deselect all **Wall Editor Tool Bar** icons.
The editor screen is in **Fine Grid** edit mode when all icons are deselected.
3. Click the **Add Tile Grid to Wall** icon to deselect it.
The editor screen reverts to **Fine Grid** edit mode. See Figure 24b.
4. Click the **Add Tile Grid to Wall** icon to select it.
The editor screen changes to **Tile Grid** edit mode. See Figure 24c.

Note: In a new wall design, there will be no tiles added to the wall.



a) Compare: **Assign Outputs**, **Tile Grid edit mode**, and **Fine Grid edit mode**



b) **Tile Grid edit mode**

c) **Fine Grid edit mode**

Figure 24 Wall Editor Edit Mode Screens

- a) Compares Assign Outputs, Tile Grid and Fine Grid Mode screens.
 b) Tile Grid edit mode.
 c) Fine Grid edit mode.

The two wall editing modes, **Tile Grid** and **Fine Grid**, are described in the following sections.

- See Section 2.6 “Wall Editor - Tile Grid Mode” on page 31.
- See Section 2.7 “Wall Editor - Fine Grid Mode” on page 46.

When designing a completely new wall, start with the **Tile Grid** mode in the **Wall Editor** to arrange tiles quickly and easily.

2.6 Wall Editor - Tile Grid Mode

Note:

In the **Wall Editor**, use the **Display Output** icons in the **Outputs Banner** to navigate to each multiviewer output:

1. Click on a **Display Output** icon to select it.
The corresponding section of the wall is selected for editing.
2. Click on another **Display Output** icon, to change section of wall.
The icons act like “radio buttons”.
3. Click on a selected **Display Output** icon to deselect it.

(See Section 2.4.2 “Outputs Banner” on page 18.)

Select the **Add Tile Grid to Wall** icon in **Wall Editor Tool Bar** for **Tile Grid** mode. See Figure 25.

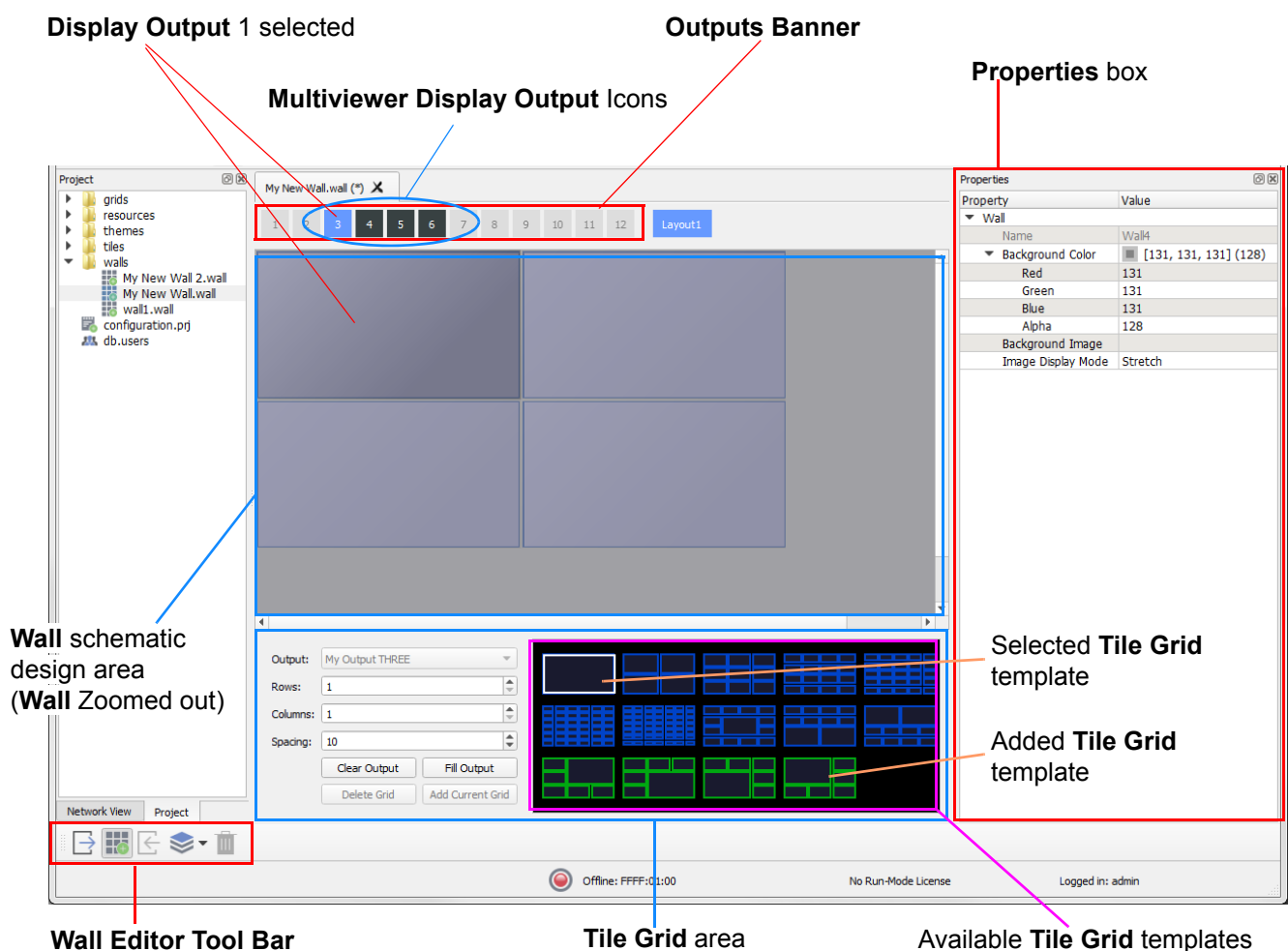


Figure 25 Wall Editor - Tile Grid Edit Mode screen

The **Tile Grid Editor** screen comprises:

- **Multiviewer Display Output** icons - used to select which multiviewer output display area of a wall is being edited.
- **Wall schematic design area** - **Tile** arrangements (**Tile Grids**) are added to this area.
- **Tile Grid** area - Various **Tile Grids** are shown and can be added to the wall. The area comprises some controls and settings, see Table 6, and an array of available **Tile Grid** templates, see Figure 26.

- **Properties Box** - Tile properties can be viewed and modified.
See Section 2.9 “Add a Custom Tile to Wall” on page 55 for more information.

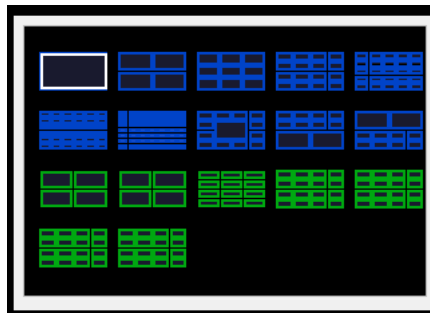
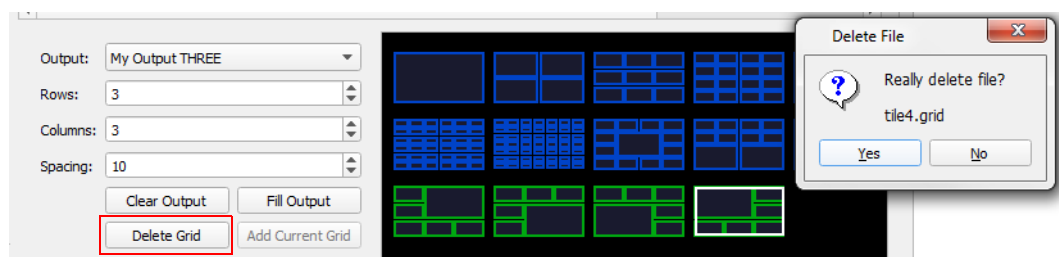


Figure 26 Tile Grid area - Tile Grid Templates

| Tile Grid Design Control / Setting | Description |
|------------------------------------|--|
| Output | Drop-box. Selects the multiviewer display output being edited. (Grayed out if a display output is already selected via a Display Output Icon.) |
| Rows | Drop-box. Select number of rows in the Tile Grid. |
| Columns | Drop-box. Select number of columns in the Tile Grid. |
| Spacing | Drop-box. Select the pixel spacing between the tiles placed on the grid. Default is 10 pixels. |
| Clear Output | Button. Click to clear tiles from the tile grid of the selected display output. |
| Fill Output | Button. Click to populate the selected display output tile grid with automatically-generated tiles (auto-tiles). |
| Delete Grid | Button. Click to delete selected Tile Grid in the Tile Grid area. |



| | |
|-------------------------|--|
| Add Current Grid | Button. Click to add the Tile Grid of the selected wall output to the Tile Grid area. |
|-------------------------|--|

Table 6 Tile Grid area - Controls and Settings.

Note: The current maximum number of tiles in a grid is 100 (**Rows** 10 and **Columns** 10),.

2.6.1 Add Tile Grid Example

For a “tile adding” example in this section, a multiviewer with four outputs will be used, with multiviewer outputs assigned to one video wall in a 2x2 arrangement:

| | |
|----------|----------|
| Output 1 | Output 2 |
| Output 3 | Output 4 |

In the following sub-sections, **Tile Grids** are added in various ways to demonstrate different Orbit **Tile Grid** facilities.

Adding a **Tile Grid** to a **Wall**:

1. Open an Orbit Project.
2. Open a Wall which has already had multiviewer display outputs assigned.
3. Deselect all the **Wall Editor Tool Bar** icons. Then select the **Add Tile Grid to Wall** icon. The **Wall Editor** is in the **Add Tile to Grid** mode.
4. Click on **Display Output** icon 1, to select it. The display area for Output 1 is focused on in the schematic editor and highlighted. See Figure 27.

Note:

To see the whole wall in the schematic editor screen: Use the **Zoom Out** (-) controls in the main tool bar to zoom out. Use the **schematic editor** area's horizontal and vertical **scroll bars** to center the whole wall.

Display Output icon 1 selected

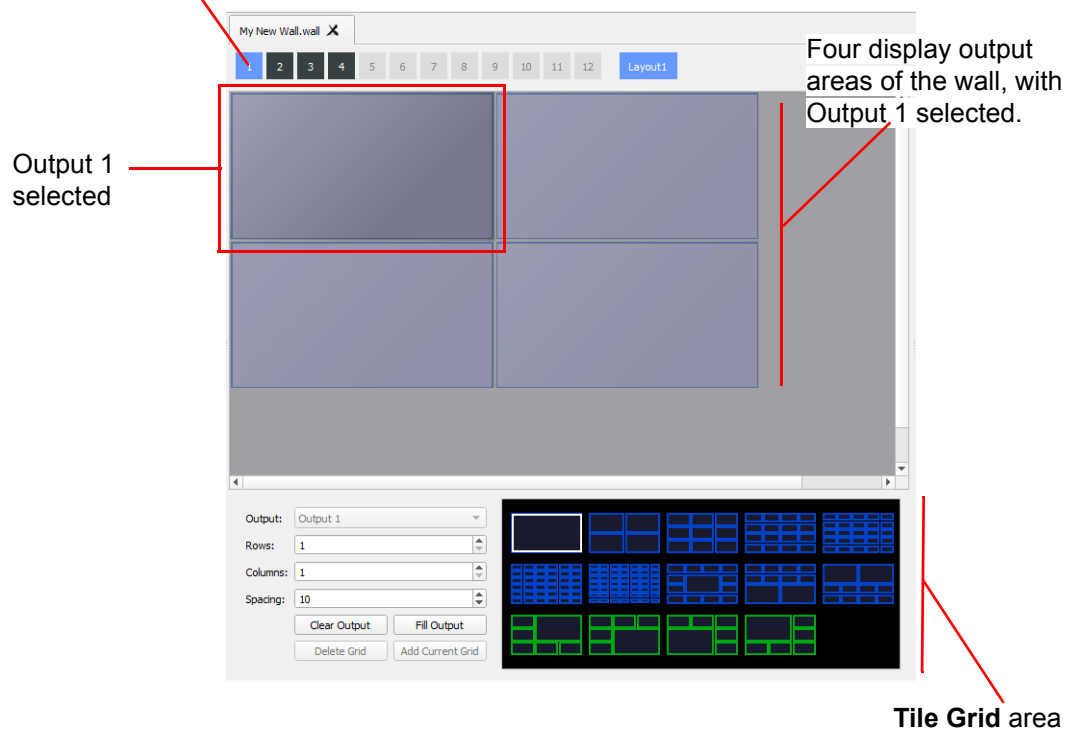


Figure 27 Output 1 selected in Wall

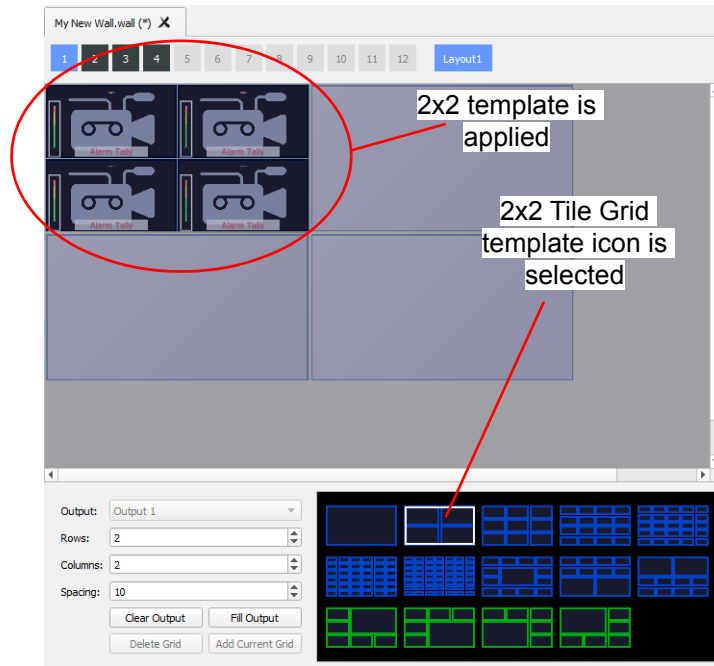


Figure 28 Output 1 with an Applied 2x2 Tile Grid

5. Click on the **Save File** icon in the main tool bar to save the change.

A **Tile Grid** has been applied to a **Display Output** on the **Wall**. The **Tile Grid** has been populated with automatically-generated video tiles (**Auto-Tiles**) which are configurable.

2.6.2 Adding a Tile Grid - Using “Output:” drop-down box

Another way to add a **Tile Grid**:

1. Ensure all **Display Output** icons are deselected.
(If required, click on a selected **Display Output** icon to deselect it.)
2. In the **Tile Grid** area,
select ‘Output 2’ in the **Output** drop-down menu. See Figure 29.

Display Output
icons all
deselected

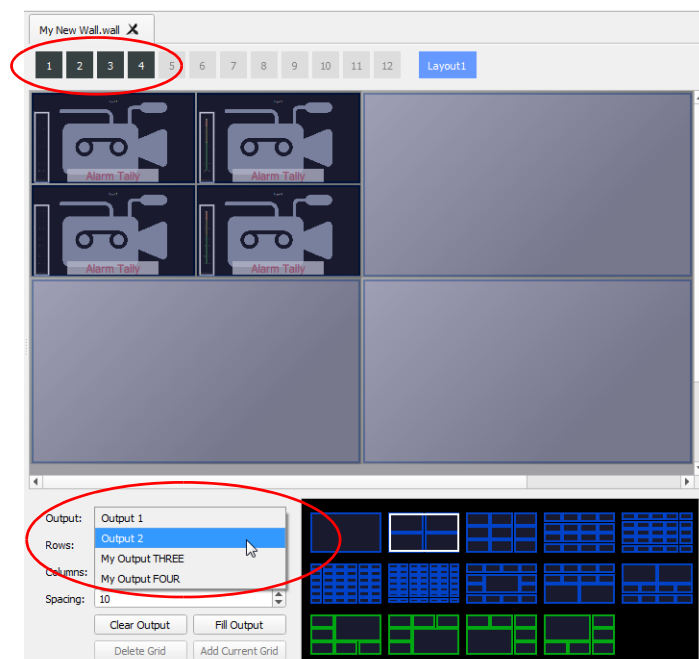
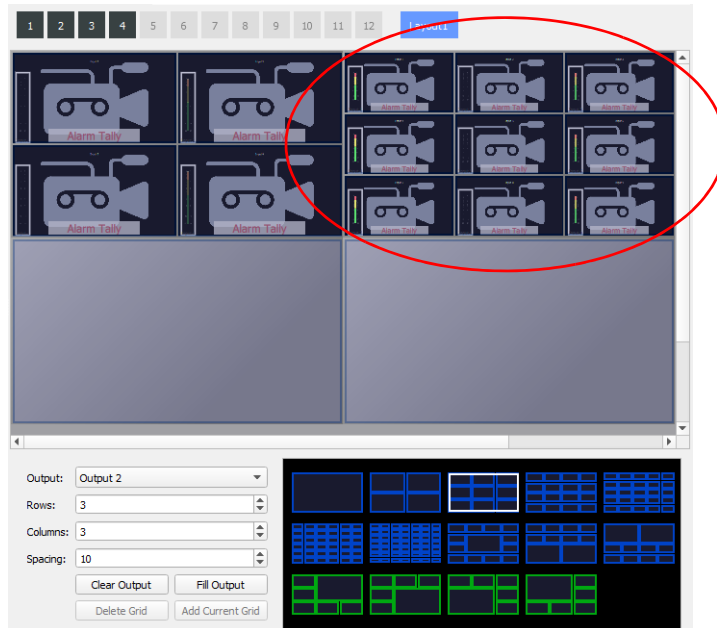


Figure 29 Selecting 'Output 2' with the "Output" drop-down

3. In the **Tile Grid** area, click on a 3x3 tile grid image.
The Output 2 display area is now a 3x3 tile grid. See Figure 30.

**Figure 30 Output 2 display area with a 3x3 Tile Grid**

4. Click on the **Save File** icon in the main tool bar to save the change.

Another **Tile Grid** has been applied to a **Display Output** on the **Wall**.

2.6.3 Adding a Tile Grid - using Grid Rows and Columns

Another way to add a **Tile Grid**:

1. Click on **Display Output** icon 3.
The display area for Output 3 is focused on in the schematic editor and highlighted.
See Figure 31.

Note:

To see the whole wall in the schematic editor screen:
Use the **Zoom Out** (-) controls in the main tool bar to zoom out and
use the **schematic editor** area's horizontal and vertical **scroll bars** to center the wall.

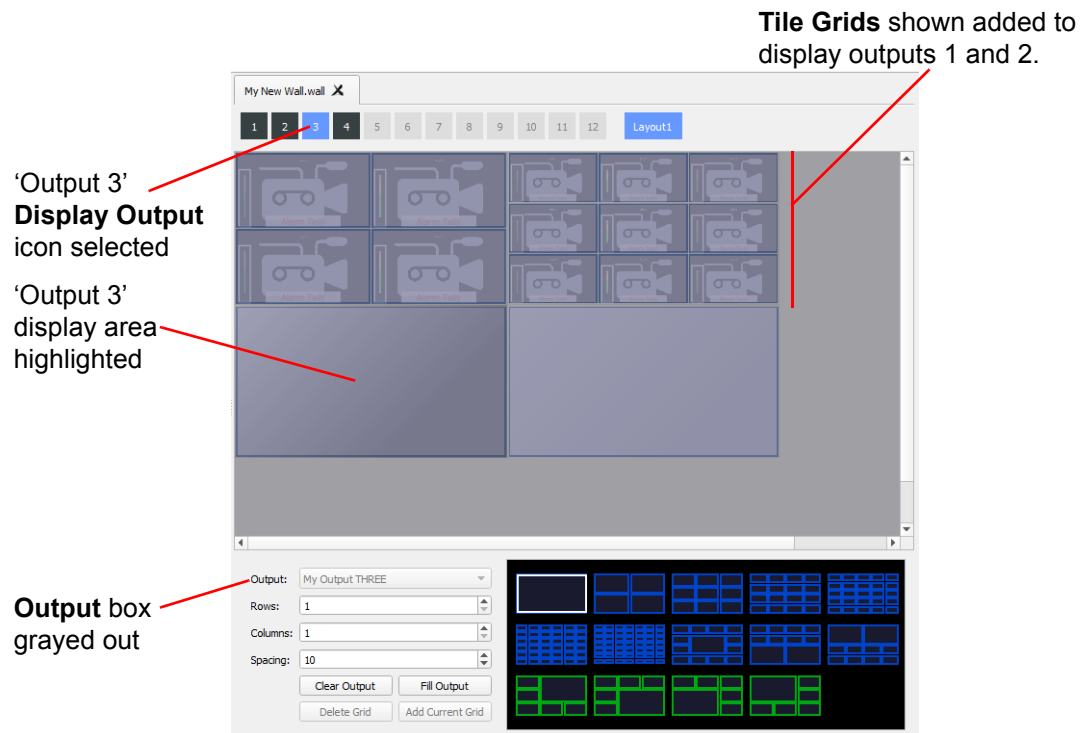


Figure 31 Selecting 'Output 3' via the Display Output icons

2. In the **Tile Grid** area, select a **Rows** value of 2 and a **Columns** value of 3.

The Output 3 display area is now a 3x2 **Tile Grid**. See Figure 32.

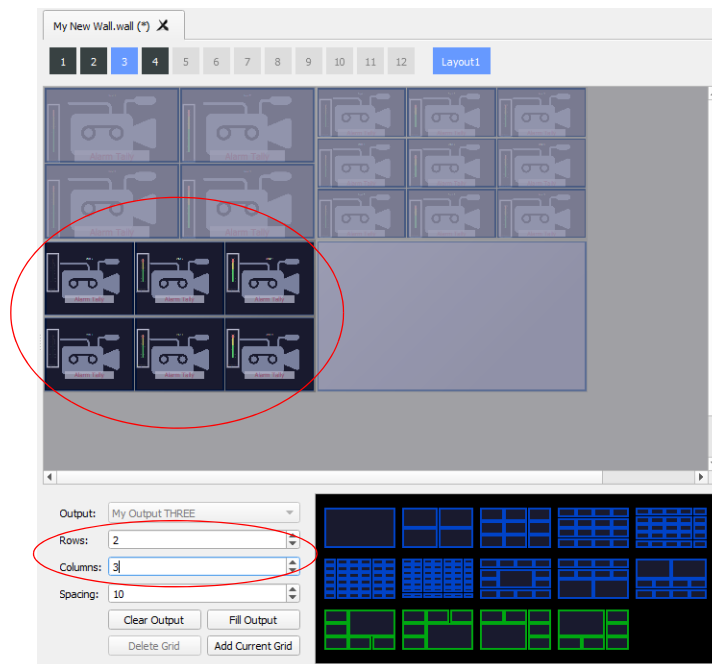


Figure 32 Output 3 display area with a 3x2 Tile Grid

3. Click on the **Save File** icon in the main tool bar to save any changes.

Another **Tile Grid** has been applied to a **Display Output** on the **Wall**.

2.6.4 Adding a Tile Grid and Adjusting Tile Spacing

1. Click on **Display Output** icon 4.
The display area for Output 4 is focused on in the schematic editor and highlighted.
2. In the **Tile Grid** area, click on a 2x2 tile grid image.
The Output 4 display area is now a 2x2 tile grid.
3. In the **Tile Grid** area, select a **Spacing** value of 150.
(This large value illustrates the effect of this setting. It is not a recommended value.)
The Output 4 display area is now a 2x2 tile grid with very wide spacing between tiles. See Figure 33.

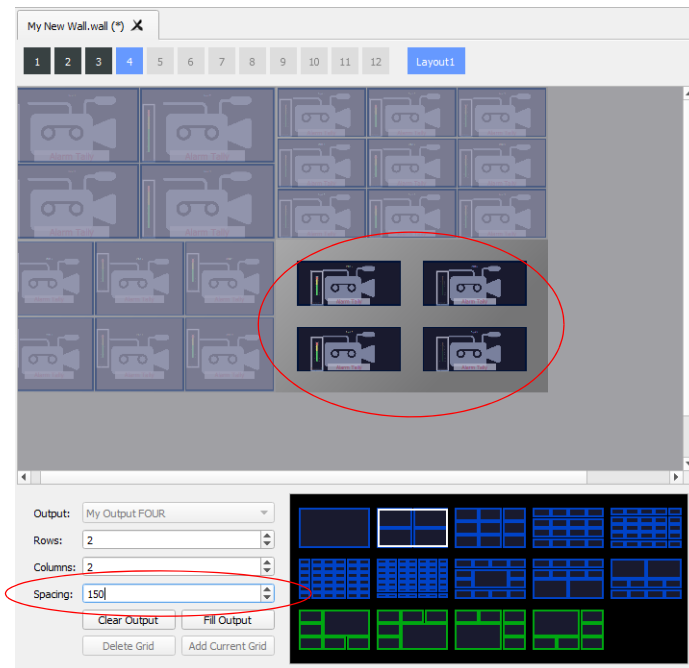


Figure 33 Output 4 display area now a 2x2 Tile Grid with a Wide Tile Spacing

Note: As the tile **Spacing** setting increases, the tile size reduces. The grid configuration is unchanged.

4. Click on the **Save File** icon in the main tool bar to save any changes.

Another **Tile Grid** has been applied to a **Display Output** on the **Wall**.

2.6.5 Resize a Tile in Tile Grid Mode

A **Tile Grid** can be adjusted on a **Display Output**.

To resize a tile on a grid:

1. Click on **Display Output** icon 1.
This shows display output 1 on the wall.
2. Use the **Scale to Fit** zoom control to see only Output 1 in the schematic area.
3. In the **Tile Grid** area, click on a 6x6 tile grid image.
The Output 1 display area is now a 6x6 tile grid.

4. Select a **Tile**, by clicking on a tile in the 6x6 grid. See Figure 34.

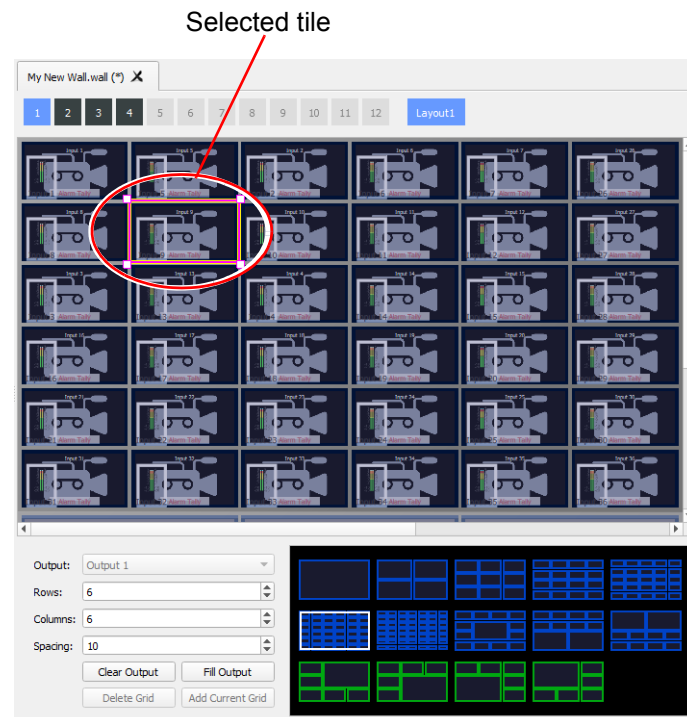


Figure 34 Tile selected in a 6x6 Tile Grid

5. Place the cursor at the lower right-hand corner of the tile:
Drag it down and to the right by three tiles. See Figure 35.
Figure 35c shows the end result, a re-sized tile, four times the size.

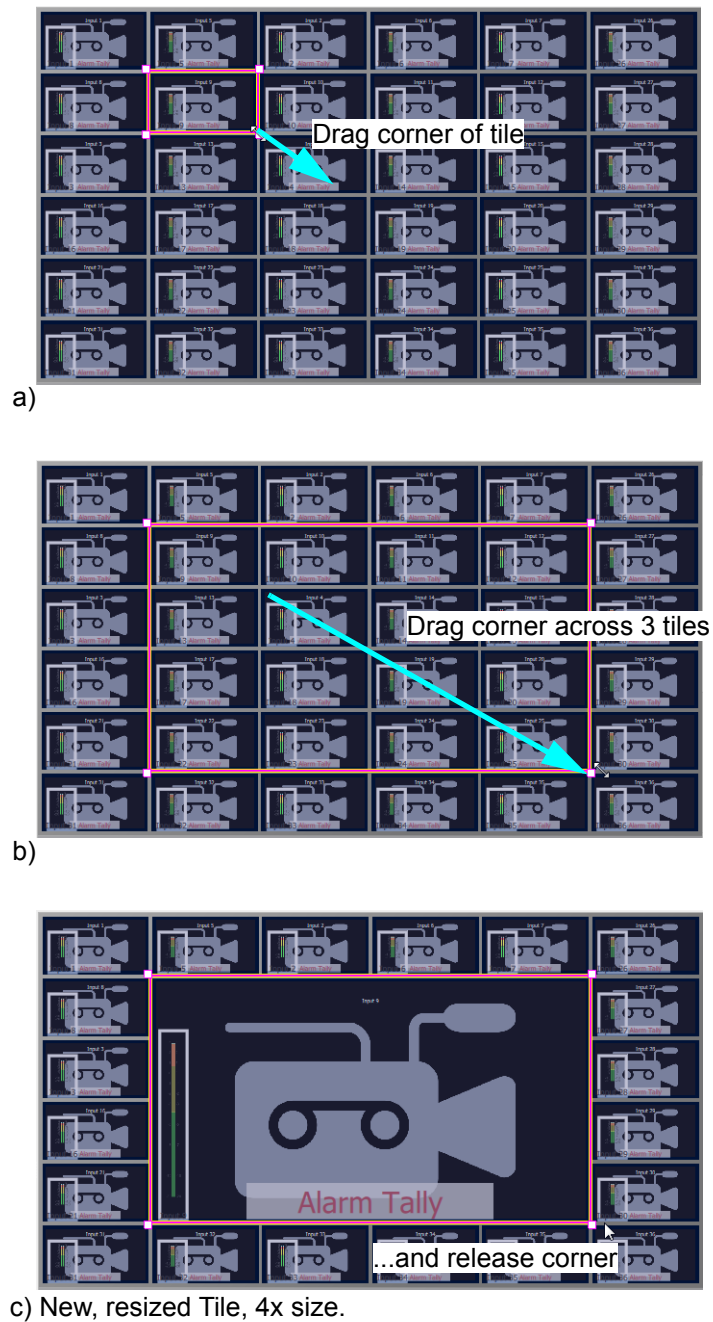


Figure 35 Resizing a tile on the Tile Grid:

- a) Drag tile corner.
- b) Drag and resize the tile.
- c) New tile, 4x size.

6. Click on the **Save File** icon in the main tool bar to save any changes.

The **Tile Grid** for the **Display Output** has been changed from a '6x6 tile' grid to one large central tile with 20 smaller surrounding tiles, a '1+20 tile' grid.

Note:

When a **Tile** is resized and enlarged, it covers any existing tiles. After resizing, those covered tiles are deleted from the **Tile Grid**.

2.6.6 Add Current Grid, '1+20 Tile' Grid

A new tile grid was created in Section 2.6.5, "Resize a Tile in Tile Grid Mode". This new grid design can be saved for further use on the project:

For example, for Output 1 from Section 2.6.5:

1. Select the **Display Output** which has a modified **Tile Grid** design.
2. Click **Add Current Grid** in the **Tile Grid Design** area.
The new '1+20 tile' grid is added to the available tile grids. See Figure 36.

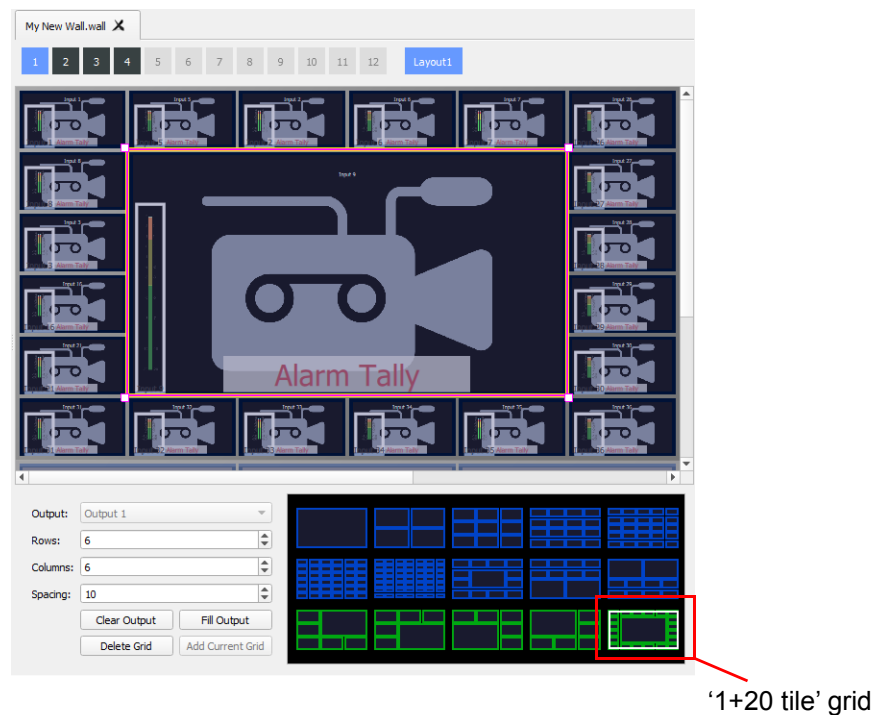


Figure 36 Add Current Grid - New Tile Grid Added to Available Tile Grids

2.6.7 Reposition a Tile in Tile Grid Mode

Again, for the example Output 1 from Section 2.6.5:

1. Select the **Display Output** which has a modified **Tile Grid** design.
2. Select the central 4x size tile.

3. Drag the tile corners to resize and reposition the tile.
See Figure 37 for an example.

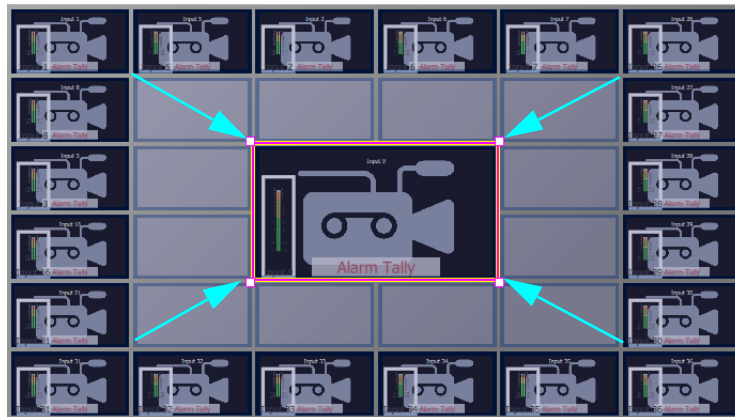


Figure 37 A Resized and Repositioned tile.

4. Click on the **Save File** icon in the main tool bar to save any changes.

Note:

In the **Tile Grid** editing mode, it is only possible to resize or reposition a tile onto corners of the (coarse) **Tile Grid** itself - a “snap to tile-grid” mode.

Fine adjustments are done in **Fine Grid** edit mode.
See Section 2.7 “Wall Editor - Fine Grid Mode” on page 46.

2.6.8 Tile Grid Mode - The Clear and Fill Output Controls

1. Click on **Display Output** icon 1.
2. In the **Tile Grid** area, click on a 6x6 tile grid image.
Or
In the **Tile Grid** area, select a **Rows** value of 6 and a **Columns** value of 6.
The Output 1 display area is now a 6x6 tile grid.
See Figure 38.

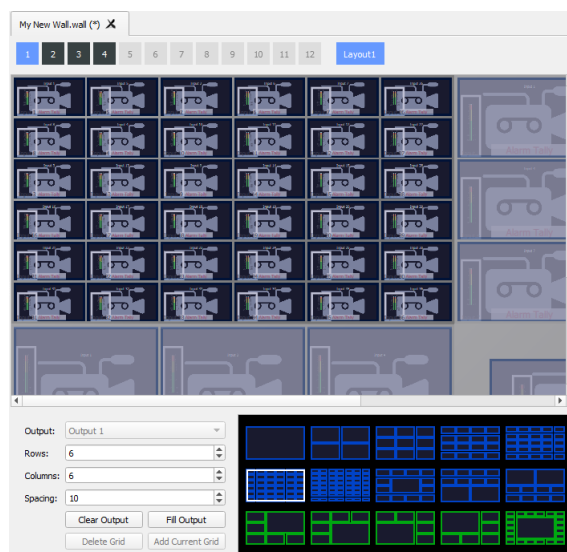


Figure 38 Output 1 display area 6x6 Tile Grid

3. In the **Tile Grid** area, click **Clear Output**.

Tiles are cleared from the tile grid (of Output 1)
and an empty 6x6 grid remains in place. See Figure 39.

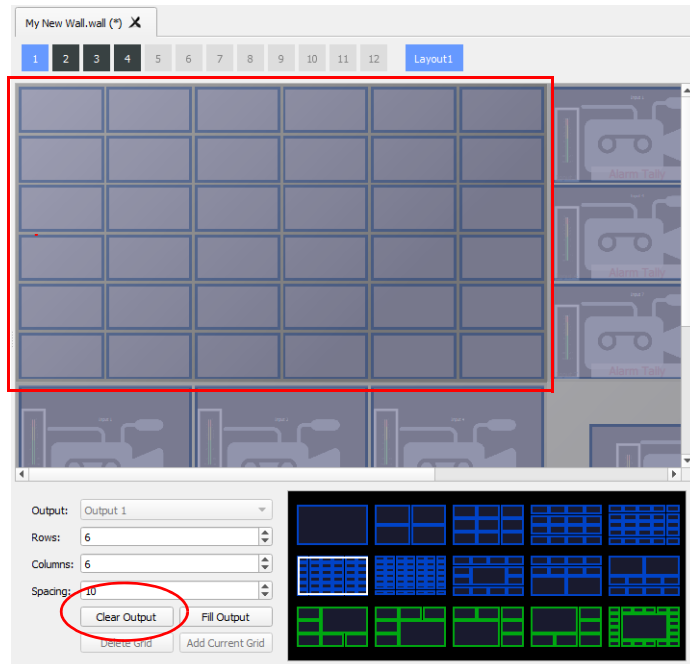


Figure 39 Cleared Output 1 - Empty 6x6 Grid Remains

4. In the **Tile Grid** area, click **Fill Output**.

The empty tile grid fills with auto-tiles. See Figure 40.

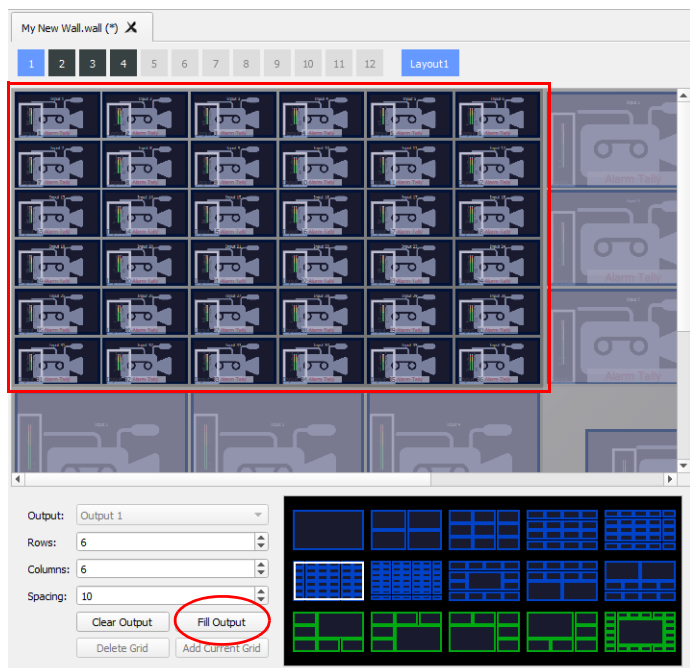


Figure 40 Output 1 display area grid Filled with auto-tiles

- Click on the **Save File** icon in the main tool bar to save any changes.

Note:**Select all Tiles:**

All tiles on the selected display output may be selected with Ctrl-A (Hold down keyboard <control> and "A").

Note:**Move Tiles:**

Selected tiles may be moved by dragging and dropping them with the cursor.

Note:

Fill Output is useful for repopulating a tile grid with auto-tiles.

Note:

Clear Output deletes all tiles from the tile grid of the selected output.

2.6.9 Tile Grid Mode - Select, Delete, Resize and Fill

Note:**Selection/ Deselection of graphical items:**

The following 'select' / 'deselect' instructions are used in the **Wall Editor** when making changes:

| Select / Deselect | Action |
|------------------------------|--|
| Deselect All | Click outside the schematic area. |
| Select | Click on an item to select it. |
| Add Select | Shift-Click on an item to add it to current selection. |
| Remove from Selection | Shift-Click on an <i>already-selected</i> item to remove it from the current selection. |
| Select All | 'Ctrl-A' selects all items. |
| Select an Area | Shift drag cursor on screen to describe a rectangular area. All objects wholly or partly in the area will be selected. |

2.6.9.1 Selecting

- Click on **Display Output** icon 1.
In the **Output Tile Grid** area, click on a 6x6 tile grid image.
The Output 1 display area is now a 6x6 tile grid.

2. To select a tile, click on a tile in the 6x6 grid. See Figure 41a.

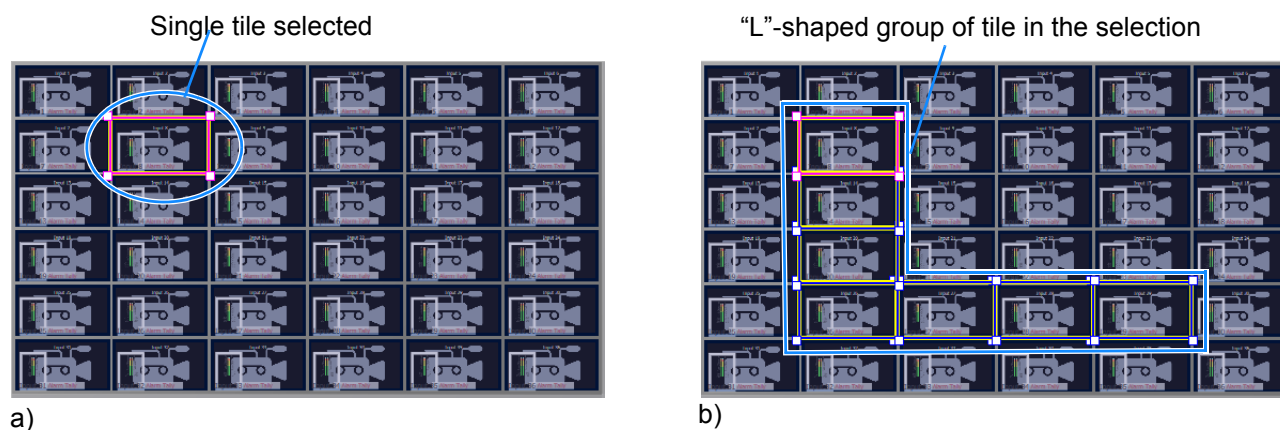


Figure 41 Tile(s) Selected in a Tile Grid:

- a) Single Tile Selected.
- b) Several Tiles Selected.

3. To select further tiles, shift-click on further tiles. (Hold down the shift key while clicking). This selects several tiles at once. See Figure 41b.
4. To deselect a selected tile, shift-click on the tile.

2.6.9.2 Deleting

5. To delete the tile(s) selected, press the 'delete' key on your keyboard.

The selected tiles are deleted and the **Tile Grid** remains in place. There are empty tile positions, "tile gaps". See Figure 42.

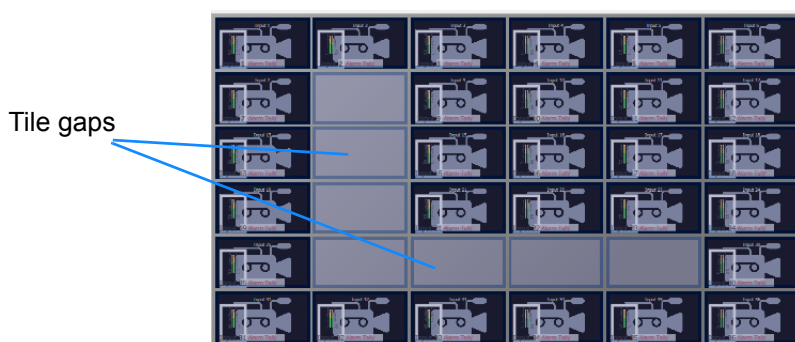


Figure 42 Deleted Tiles in a 6x6 Tile Grid

Note:

Undo and Re-do:

After deletion, Ctrl-Z on the keyboard will **undo** the delete operation.

Ctrl-Z performs an "undo" function.

Ctrl-Y performs a "redo".

2.6.9.3 Filling

6. Resize a tile in Output 1, as shown in Figure 43.

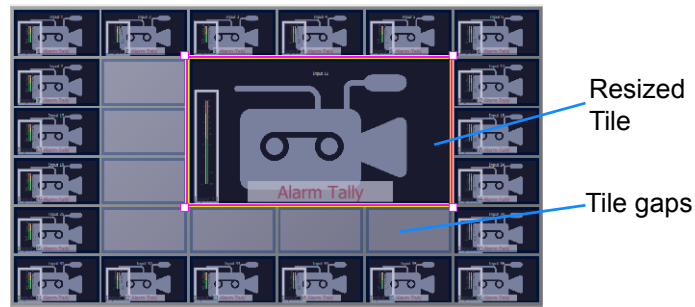


Figure 43 Resized tile

7. In the **Tile Grid** area, click **Fill Output**.

The empty tile gaps are filled with tiles.

The resized tile is unaffected. See Figure 44.

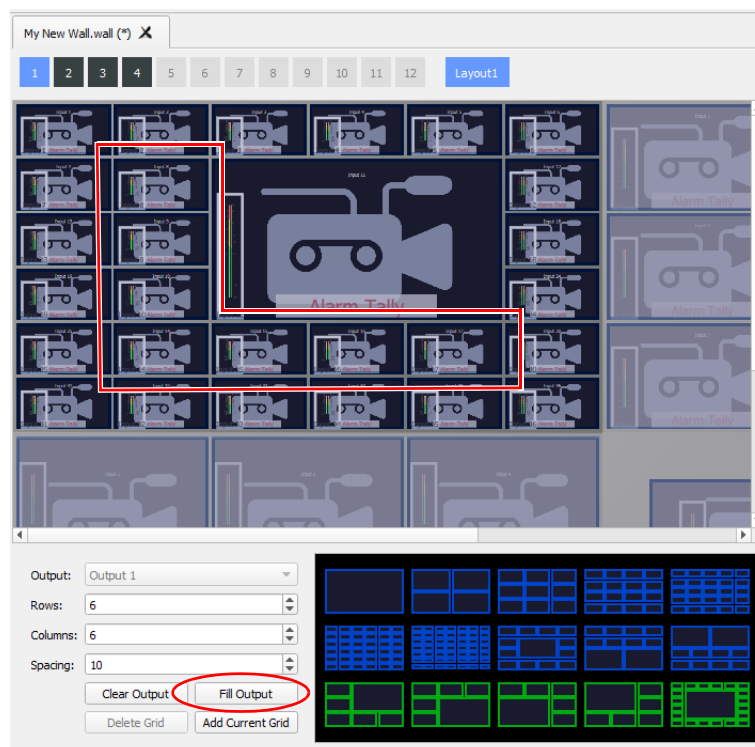


Figure 44 Fill Output Fills the “tile gaps”

8. Click on the **Save File** icon in the main tool bar to save any changes.

Note: **Fill Output** fills out a display output according to the **Output**, **Rows**, **Columns** and **Spacing** settings in the **Tile Grid** area.

Note: **Fill Output** populates empty grid positions with a video **Auto-Tile** type of tile.

This is generated by Orbit with some default tile properties, i.e. a generic video tile with audio bars on the left-hand side. The properties may be edited.

2.7 Wall Editor - Fine Grid Mode

Note:

In the **Wall Editor**, use the **Display Output** icons in the **Outputs Banner** to navigate to each multiviewer output:

1. Click on a **Display Output** icon to select it.
The corresponding section of the wall is selected for editing.
2. Click on another **Display Output** icon, to change section of wall.
The icons act like “radio buttons”.
3. Click on a selected **Display Output** icon to deselect it.

(See Section 2.4.2 “Outputs Banner” on page 18.)

When a grid of tiles has been placed onto a wall design (via in the **Tile Grid edit mode**), then fine adjustment of tile position and size may be done in the **Fine Grid** edit mode.

The grid used in the schematic area in **Fine Grid** mode is *not* the coarse “**Tile Grid**” but, instead, it is a finer, user-defined grid, the **Fine Grid**.

For the **Fine Grid**, the following apply:

- Its grid size is set up in the **Tools > Options > Grid** menu.
- It can be made visible or invisible.
- It can be shown in front of or behind the graphical objects being edited.

To open the **Wall Editor** in **Fine Grid** mode:

1. Open a wall in the **Wall Editor**.
2. In the **Wall Editor Tool Bar**, *deselect* both the **Assign Outputs to Wall** and the **Add Tile Grid to Wall** icons.
The tile **Fine Grid** mode Wall Editor screen is then shown. See Figure 45.

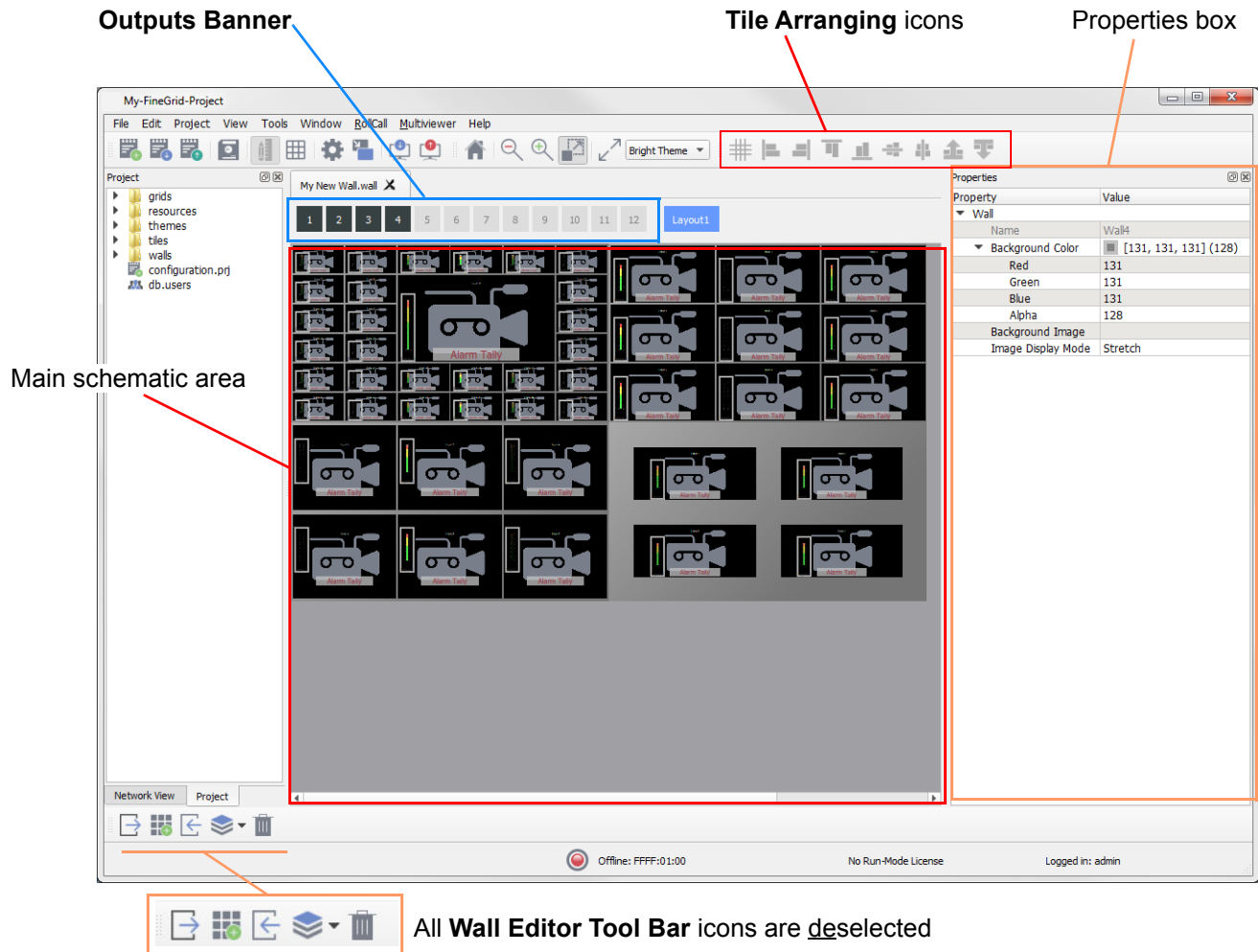


Figure 45 Wall Editor - Fine Grid Mode Screen

The **Fine Grid** mode editor screen comprises:

- **Outputs Banner** - used to select which part of the wall (i.e. which multiviewer output) is being edited.
See Section 2.4.2 "Outputs Banner" on page 18 for more information.
- **Schematic Area** - Tile sizes and positions are modified in this area.
- **Tile Arranging** icons - Tile arranging tools for aligning, distributing and front/back layering tiles placed on a wall. These are located in the Orbit screen main tool bar.
- **Properties Box** - Allows tile properties to be viewed and modified.
See Section 2.9 "Add a Custom Tile to Wall" on page 55 for more information.

2.7.1 Arranging Tiles

The **Tile Arranging** tools form a group of icons in the **Fine Grid** editor, see Figure 46. The icons are grayed out variously when tiles are not selected in the wall schematic. Icon functions are described in Table 7.



Figure 46 Tile Arranging icons

- a) Tile Arranging Icons in the tool bar, normally grayed out.
 b) Icons active when a single tile is selected.
 c) Icons active when more than one tile is selected.

| Icon | Description |
|----------------------|---|
| Align as Grid | Click to place selected tiles in an array arrangement. See "Align as Grid Tool" on page 49. |
| Align Lefts | Click to align left-hand sides of selected tiles. Alignment is made to the first tile selected. |
| Align Rights | Click to align right-hand sides of selected tiles. Alignment is made to the first tile selected. |
| Align Tops | Click to align top edges of selected tiles. Alignment is made to the first tile selected. |
| Align Bottoms | Click to align bottom edges of selected tiles. Alignment is made to the first tile selected. |
| Align Middles | Click to horizontally align middles of selected tiles. Used in Wall or Tile Editor screen. Alignment is made to the first tile to have been selected. |
| Align Centers | Click to vertically align bottom centers of selected tiles. Used in Wall or Tile Editor screen. Alignment is made to the first tile to have been selected. |
| Move to Front | Click to bring selected tile to the front in the Z order. (Top layer.) |
| Move to Back | Click to move selected tile to the back in the Z order. (Bottom layer.) |

Table 7 Tile Arranging icons

The **Tile Arranging** icon group can be undocked from the Orbit main bar and docked elsewhere or be left floating.

To undock:

- Place the cursor on the icon group's **Move Handle**, drag the icon group and release.

To re-dock:

- Drag the icon group back to the main tool bar.

(See the 'Orbit - Introduction User Manual' for information on docking/undocking the main tool bar.)

2.7.1.1 Align as Grid Tool

The **Align as Grid** tool places selected tiles into a regular array arrangement on the wall. The example below shows how a haphazard tile arrangement may be arranged uniformly:

- Open a **Wall** containing some tiles and enter **Fine Grid** mode.

Note:

To toggle visibility of the fine grid *behind* the tiles, click the **Grid** icon in the Orbit main tool bar or click **View > Grid**.

- Select some tiles in the wall editor screen for rearranging. For example, see Figure 47.

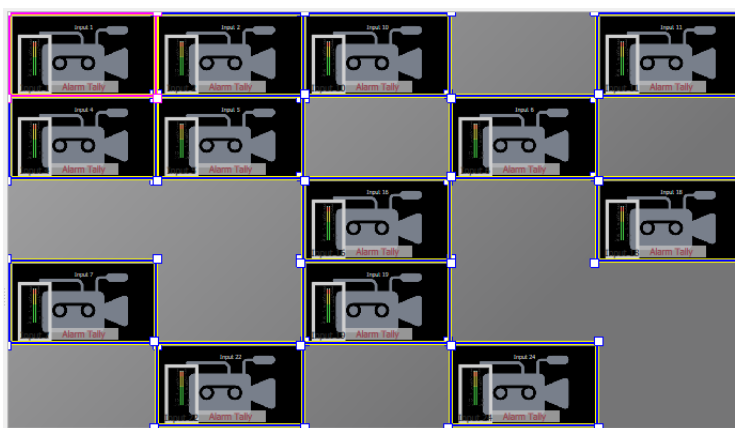


Figure 47 A Selection of Haphazardly-arranged Tiles.

- Click on the **Align on Grid** icon. The user is prompted for some grid array parameters in the properties area. See Figure 48 and Table 8.

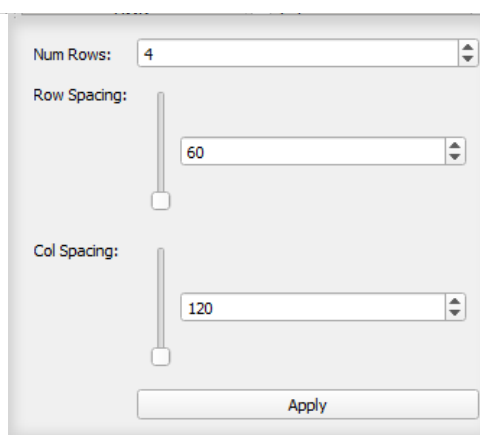


Figure 48 Align To Grid Array Parameters

| Parameter | Description |
|--------------------|--|
| Num Rows | Enter the number of rows of tiles in the array. |
| Row Spacing | Enter the number of pixels between the rows of tiles. |
| Col Spacing | Enter the number of pixels between columns of tiles. |
| Apply | Button. Click to apply the Align to Grid parameters to the tile selection. |

Table 8 Align to Grid Parameters

4. Enter the following parameters for this example:

- **Num Rows** = 4.
- **Row Spacing** = 60.
- **Col Spacing** = 120.

Click **Apply**.

The selected tiles are placed on screen in an array arrangement, see Figure 49. The tiles are placed on screen filling each column of the array in turn to form the required number of rows, starting from the left.

Note:

If the result of performing an **Align as Grid** operation places tiles off-screen, **Ctrl-Z** will undo the operation.

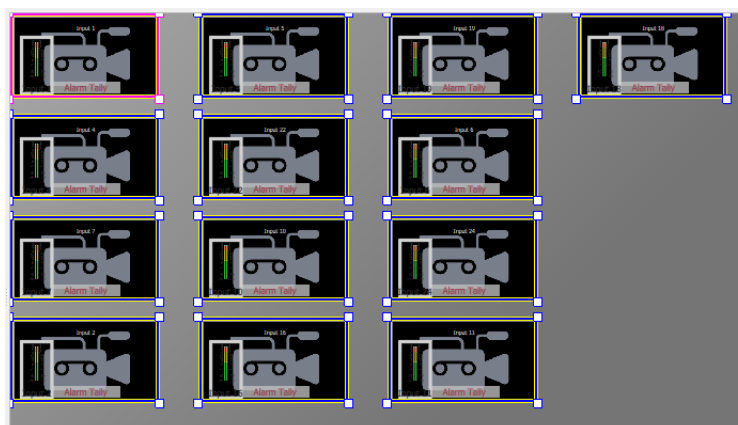


Figure 49 Tiles Aligned To Grid

2.8 Fine Adjustment of Tiles on a Wall

Fine adjustment of tile size and position can:

- Enable a different tile grid to be achieved.
For example, a desired tile arrangement may not have regular, uniform tile spacing.
- Allow slight adjustment to tile positions on a video wall.
For example, to create space on the video wall for:
 - A wall title to be squeezed in.
 - A small logo or clock.

This section shows how to do fine adjustments to tile size and position in the **Wall Editor** in **Fine Grid** mode.

1. Open **Wall** design with a 4x4 **Tile Grid**.
2. Go to the **Fine Grid** wall editor screen and make the fine grid visible. See Figure 50.

Note:

Fine Grid visibility:

To toggle visibility of the fine grid behind the tiles, click the **Grid** icon in the Orbit main tool bar or click **View > Grid**.

Note:

For **Fine Grid** mode, ensure the **Add Tile Grid to Wall** icon in the **Wall Editor Tool Bar** is deselected.



Arranging Tiles icons (see Section 2.7.1 “Arranging Tiles” on page 48)

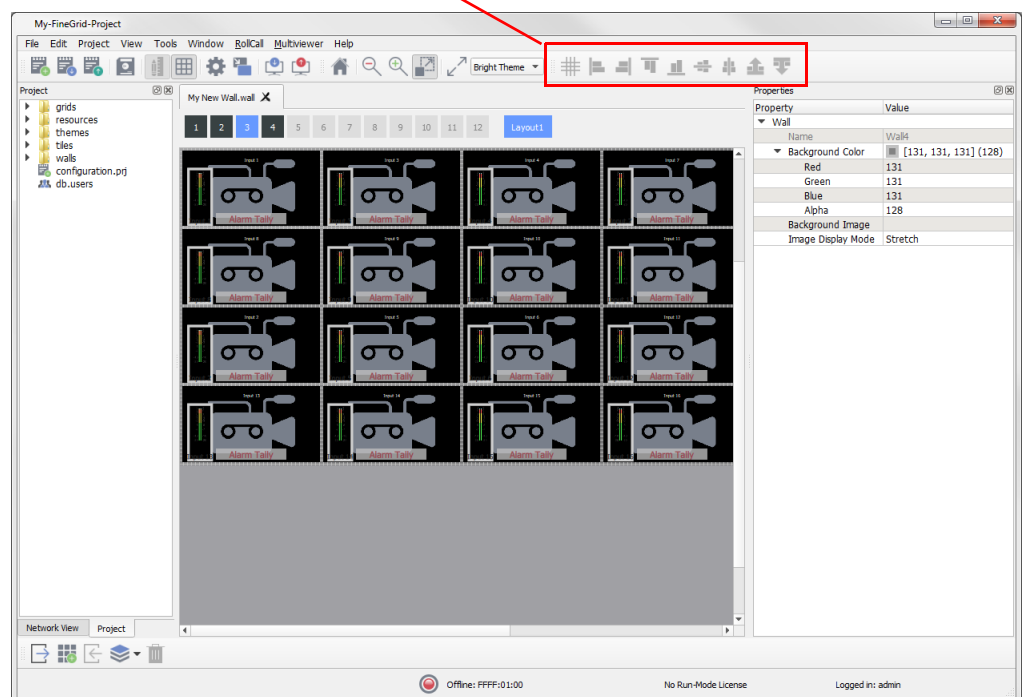


Figure 50 Fine Grid Mode Screen with 4x4 Tiles on an Output

3. Click on the **Options** icon in the Orbit main tool bar or click **Tools > Options**.
The **Tools > Options** tab is displayed. See Figure 51.
4. In the **Grid** tab,
set a grid size of 16 and select **Snap to Grid**. See Figure 51.
Click **Close**.

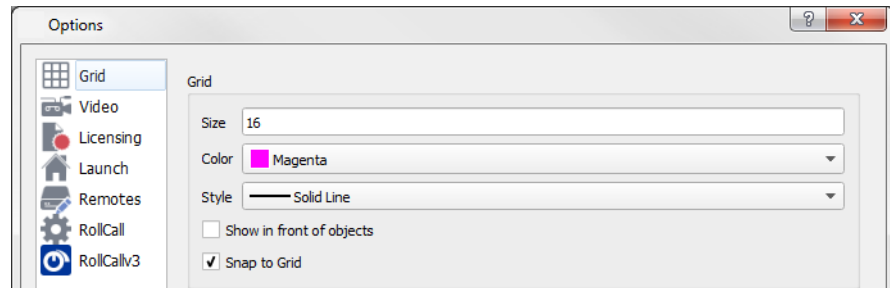



Figure 51 Tools > Options - Grid Tab

5. To toggle the **Grid** visibility on/off, click the **Grid** icon in the main menu.  .
Set the **Grid** to be visible.

This has set up the **Fine Grid**.

Note: To view the required tile on the schematic area, use the **Zoom** and **Pan / Scroll** controls.

2.8.1 Fine Position and Size Adjustment

6. Select a **Tile** on the **Wall** by clicking on it.
7. Move the **Tile** by dragging it.
8. Resize the **Tile** by dragging its corners.

The position and size of each tile may be finely adjusted in this way. Fine adjustment control can be achieved by using your keyboard's arrow keys. Figure 52 shows an example re-sized and re-positioned tile.

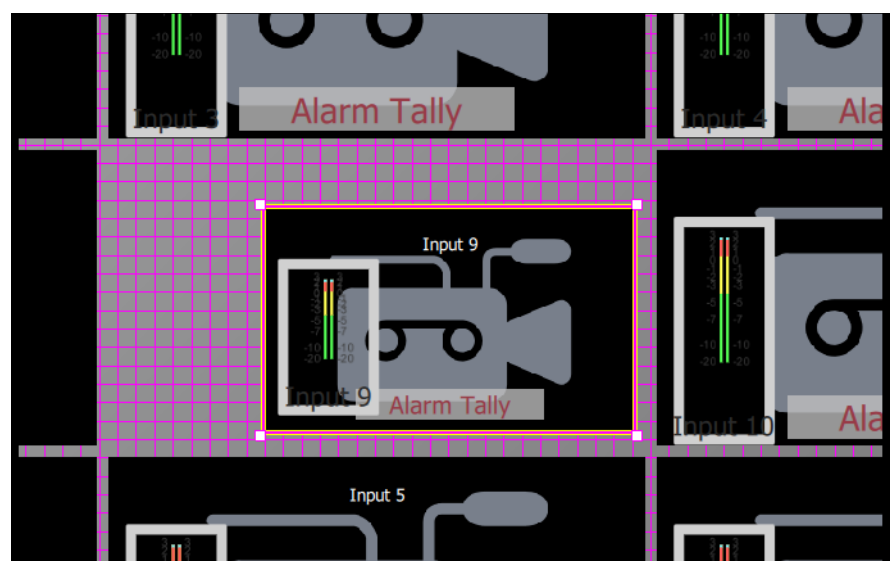




Figure 52 Tile Size and Position finely-adjusted

Remember to click the **Save File** icon in the Orbit main tool bar to save any tile layout changes.

2.8.2 Layering (Z order) of Tiles

The tile may also be place in front of or to the back of other **Tiles**:

9. Select a Tile.

10. Click on the **Move to Front**, , and the **Move to Back**, , icons to move the **Tile** in front of or behind other **Tiles**.

Note:

Fine Grid mode versus Tile Grid mode - Tile Size and Position Adjustment:

In **Fine Grid** mode, tile size and position can be finely-adjusted.

In **Tile Grid** mode, tiles can be quickly arranged on the screen, i.e. onto a tile grid: A **Spacing** control permits gaps between **Tiles**, which can result in a **Tile** being positioned off the **Tile-Grid**. See Figure 53.

I.e. **Tiles** located with a **Tile Grid** are in regular, uniform positions and the tiles themselves may be offset from the - coarse -**Tile Grid** by any tile-spacing value that has been used.

As a result, the tiles themselves may not be exactly on the **Tile Grid** itself but offset from it.

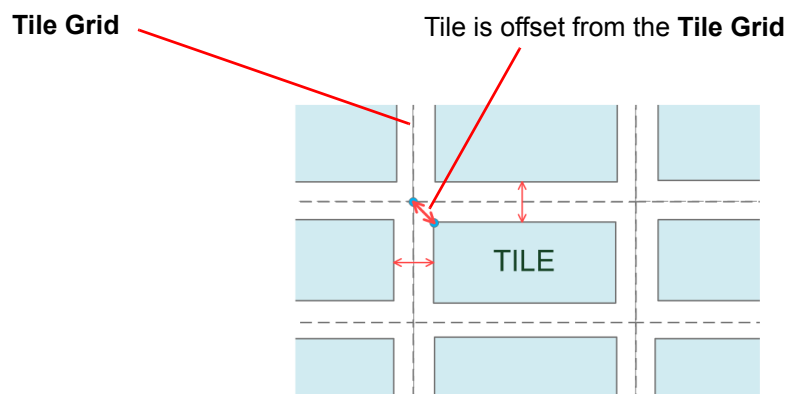


Figure 53 Tile offset from the Tile Grid

Note:

Grid options have a 'snap to grid' setting, which can be enabled or disabled to suit. This snaps to the fine grid in **Fine Grid** mode.

Note:

Caution:

After **Fine Grid** editing of tile layouts, do not use **Fill Output** in the **Tile Grid** edit mode. Because finely-resized and finely-positioned tiles can be deleted using **Fill Output**.

This is because finely-resized and finely-positioned tiles are most likely not located exactly on a coarse **Tile Grid**. The **Fill Output** control adds auto-tiles to empty coarse **Tile Grid** locations and deletes all other tiles.

2.8.3 Fine Adjustment Example:

In the on-screen example of Figure 54, **Tiles** in the central two columns have each been reduced in size vertically and moved down the screen in **Fine Grid** mode. This has created space across the top of the columns for a wide tile., For example, a text label to be a wall title.

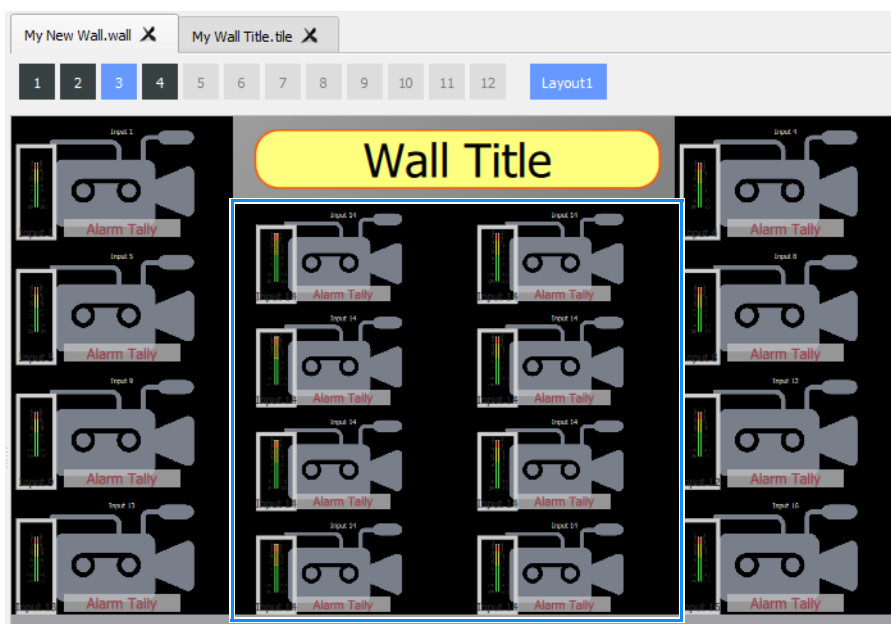


Figure 54 Example result of some Fine Adjustment of Tile sizes and positions

2.9 Add a Custom Tile to Wall

Custom Tiles may be created by the user. **Custom Tile** creation is described in Section 3 “Tile Editor - Tiles and Widgets” on page 73.

Individual **Custom Tiles** may be added to a **Wall** in the **Wall Editor** in the **Wall Editor’s Tile Grid** or in the **Fine Grid** screen.

1. In the **Project View**, expand the ‘tiles’ sub-folder to show all the **Custom Tile** designs (.tile files).
2. Click on a tile design to select it, see Figure 55.

Custom Tiles

Custom Tile Selected

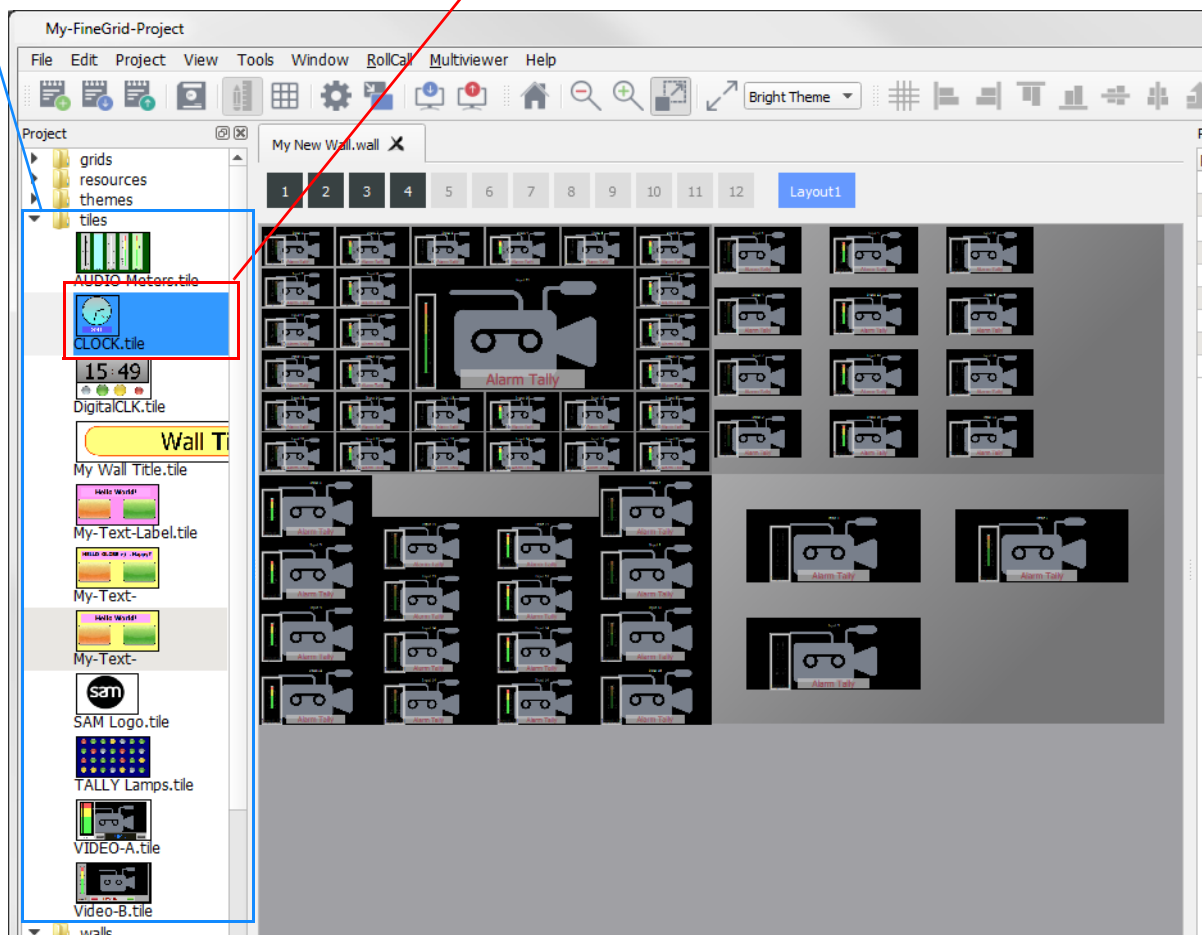
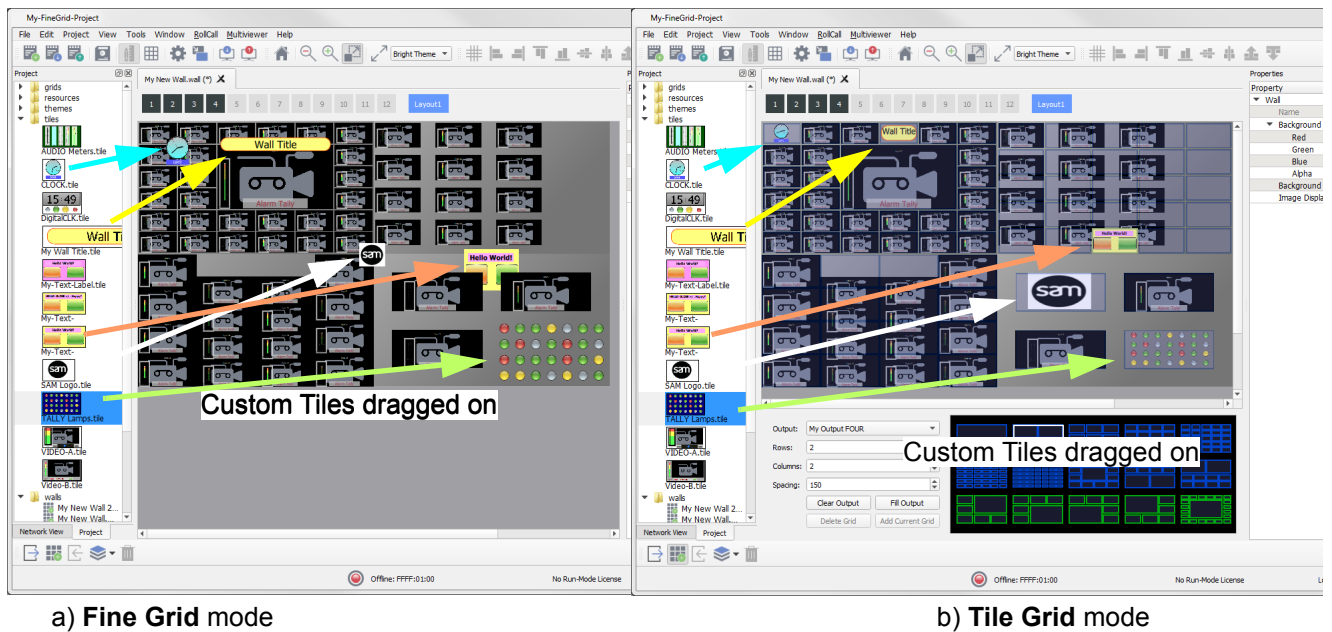


Figure 55 Custom Tile Design Selected in Project View

3. Click and drag the selected **Custom Tile** design onto the wall and release where you wish to place the **Tile**.
The **Tile** appears on the **Wall**.

In **Fine Grid** mode,
the top left-hand corner of the dragged-on tile is at the drop position. See Figure 56a.

In **Tile Grid** mode,
the custom tile replaces the tile at the drop position. See Figure 56b.



a) Fine Grid mode

b) Tile Grid mode

Figure 56 Custom Tiles Dragged on Wall:

a) Fine Grid Mode.

b) Tile Grid Mode.

2.10 Wall Tile Properties

In the **Wall Editor** schematic screens, the **Properties Box** displays properties of selected graphical items, for examples, multiviewer outputs, walls and tiles.

Properties can be viewed and modified with the **Properties Box**.

- Properties of the **Tiles** on the Wall are described in this sub-section.
- Properties of the multiviewer's display outputs are described in Section 2.4.4 "Display Output Properties" on page 21
- Properties of the **Wall** are described in Section 2.4.5 "Display Wall Properties" on page 22.

Note:

The **Tile** properties presented depend on context.
For example, in the **Wall Editor** screen:

- In **Fine Grid** mode, tile position properties may be changed.

whereas,

- In **Tile Grid** mode, these properties are shown grayed out and may not be changed.

In the **Wall Editor** screens, at the 'wall-level', the properties of a **Tile** can be modified and are unique to that selected **Tile** in the wall. Thus:

- Automatically-generated tiles can have different properties depending on wall position.
- Changes made to custom tiles at *wall-level* will apply specifically to tiles in certain wall positions.
- Changes made to custom tiles at *tile-level* will apply across all custom tiles of the same name.
- Widget Styles on a tile-by-tile basis in a wall can be selected.

To see the wall Tile Properties, select a **Tile** in a **Wall**. See Figure 57. and Table 9.

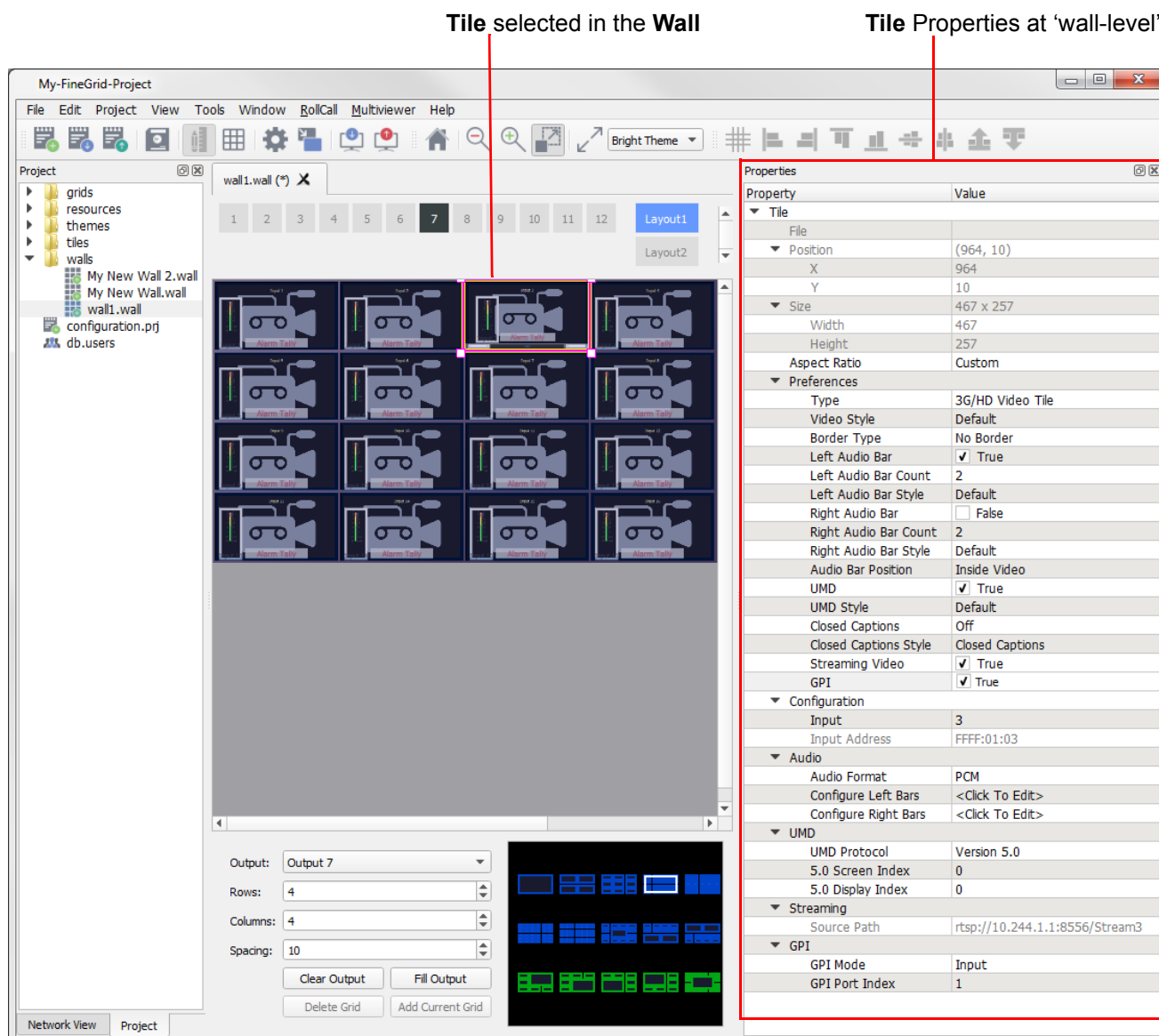


Figure 57 Wall - Video Tile Properties

| Property | Description |
|------------------|--|
| File | This is blank for an Auto-Tile . For a Custom Tile , this is the Orbit project file name for the Custom Tile . Note: If the Tile is an automatically-generated Tile , then there is no associated file name and the property value is 'n/a'. |
| Position: | (X, Y) |
| X | Horizontal position of top left-hand corner of tile relative to top left-hand corner of video wall. (Pixels.) |
| Y | Vertical position of top left-hand corner of tile relative to top left-hand corner of video wall. (Lines.) |
| Size: | (Width, Height) (Grayed out in Tile Grid mode) |
| Width | Width of the tile. (Pixels.) |

Table 9 Video Auto-Tile Properties, in Wall Editor screen

| Property | Description |
|------------------------------|---|
| Height | Height of the tile. (Lines.) |
| Aspect Ratio | Down-down box. Select tile aspect ratio: 4:3, 14:9, 16:9, and Custom. |
| Preferences: | |
| Type | <p>Drop-down box. Tile type. Click on property value to get a drop-down menu list of the automatically-generated tile types available (Auto-Tile types):</p> <ul style="list-style-type: none"> • 3G/HD Video Tile. • 4K Video Tile. • Analogue Clock. • Digital Clock. <p>Note: This list of Auto-Tile types does not include any custom-built tiles. Custom Tiles may be dragged onto the tile grid from the Project View's 'tiles' sub-folder.</p> |
| Video Style | Drop-down box. Select Video widget style. Styles created in the Themes Editor appear listed. |
| Border Type | Drop-down box. Select border for tile. |
| Left Audio Bar | Check box. True/False. Audio bars screen-position parameter. Select to place audio bars on left-hand side of tile. |
| Left Audio Bar Count | Number of audio bars to appear on left-hand side. |
| Left Audio Bar Style | Drop-down box. Select Audio widget style for the left-hand-side widget. |
| Right Audio Bar | Check box. True/False. Audio bars screen-position parameter. Select to place audio bars on right-hand side of tile. |
| Right Audio Bar Count | Number of audio bars to appear on right-hand side. |
| Right Audio Bar Style | Drop-down box. Select Audio widget style for the right-hand-side widget. |
| Audio Bar Position | <p>Drop-down box. Audio bars screen-position parameter. Audio bars may be positioned:</p> <ul style="list-style-type: none"> • Outside of the video. <p>or</p> <ul style="list-style-type: none"> • Inside (within) the video. |
| UMD | <p>Check box. True/False. Select to add an Under Monitor Display (UMD) to the Auto-Tile.</p> <p>When selected, additional UMD properties appear in the properties box (see below in this table).</p> |
| UMD Style | Drop-down box. Select UMD widget style. Styles created in the Themes Editor appear listed. |
| Closed Captions | Drop-down box: Controls display of closed caption information on widget. Off, Bottom, Middle, and Top. |

Table 9 Video Auto-Tile Properties, in Wall Editor screen (Continued)

| Property | Description |
|------------------------------|---|
| Closed Captions Style | <p>Drop-down box.</p> <p>Select style of the on-screen closed captions text.</p> <p>Default, Closed Captions, and Field.</p> <p>Styles created in the Themes Editor appear listed.</p> |
| Streaming Video | <p>Check box. True/False.</p> <p>Select to connect the tile to a H.264 video stream, for example from a Grass Valley MV-8 series multiviewer.</p> <p>Video is then displayed in the tile when in Orbit's Run Mode.</p> <p>This property applies to Orbit applications with a MV-800-DT license.</p> <p>When selected, additional Streaming Video properties appear in the properties box (see below in this table).</p> <p>Note: A MV-8 series multiviewer unit can output H.264-compressed video streams of each of its multiviewer inputs.</p> <p>Thus, a multiviewer may be the source of H.264 streaming video.</p> <p>The multiviewer's IP address and Port number are set up with RollCall Control Panel.</p> |
| GPI | <p>Check box. True/False.</p> <p>Select to associate the tile with a GPIO (GPI or GPO) of the target Grass Valley MV-8 series multiviewer.</p> <p>When selected, additional GPI properties appear in the properties box (see below in this table).</p> <p>Note: The physical GPIO of the target multiviewer are set up in the RollCall Control Panel GPI template screen.</p> |
| Configuration: | |
| Input | <p>Multiviewer input number connected to the tile.</p> <p>This is unique to each instance of a tile on a wall.</p> <p>Property is typically set at <i>tile-level</i> to be: {Input}</p> <p>This property value is expanded at <i>wall-level</i> to assign multiviewer input numbers to tiles.</p> <p>Note: Property values in braces { } indicate an Orbit project variable. Here the variable name is 'Input'.</p> |
| Input Address | Grayed out. Displays the RollCall address of the tile. |
| Audio: | |
| Audio Format | <p>Drop-down box.</p> <p>Set up the audio format being received by the tile.</p> <p>Choices:</p> <ul style="list-style-type: none"> • PCM - Normal uncompressed audio. • Dolby - Dolby-compressed audio. • Auto - Automatic detection of audio format. |

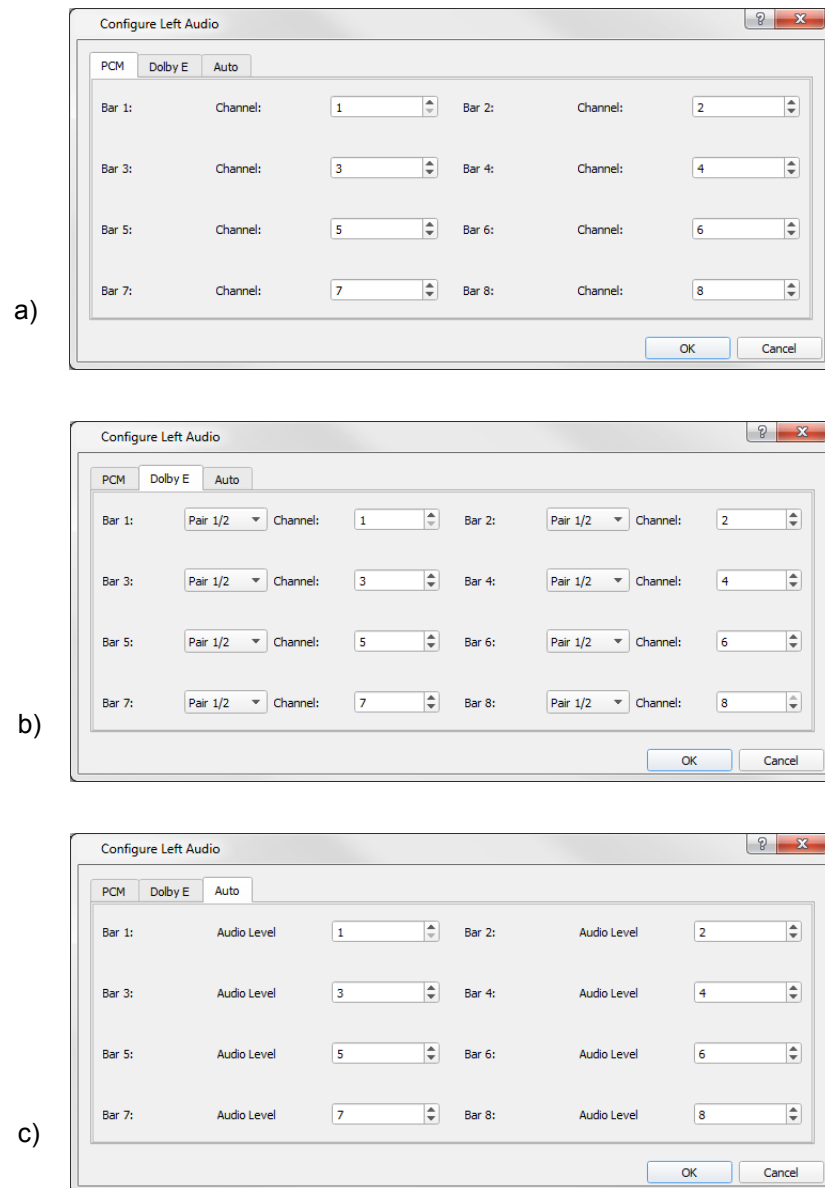
Table 9 Video Auto-Tile Properties, in Wall Editor screen (Continued)

| Property | Description |
|-----------------------------|--|
| Configure Left Bars | <p>Only available if Left Audio Bar property is selected.</p> <p>Click the value to show a Configure Left Bars dialog. See Figure 58. The dialog contains the following properties, which may be set up for each audio bar:</p> <p>PCM tab: See Figure 58a.</p> <ul style="list-style-type: none"> Channel - Select audio channel for each audio bar. <p>DolbyE tab: See Figure 58b.</p> <ul style="list-style-type: none"> Select audio channel pair (pair 1/2, 3/4 etc) in drop-box. Channel - Select audio channel. <p>Auto tab: See Figure 58c.</p> <ul style="list-style-type: none"> Select Audio Level. <p>Click OK when changes are all done in the Configure Left Bars dialog.</p> |
| Configure Right Bars | <p>Only available if Right Audio Bar property is selected.</p> <p>Click the value to show a Configure Right Bars dialog. (Similar to Configure Left Bars.)</p> |
| UMD: | Additional UMD Properties. |
| UMD Protocol | <p>Drop-down box.</p> <p>Select UMD protocol to use:</p> <ul style="list-style-type: none"> TSL Version 3.1 TSL Version 5.0 SWP-08 (Grass Valley protocol, used to get Tally names) |
| 3.1 Display Address | <p><i>Property shown for TSL version 3.1.</i></p> <p>Address to identify display, video signal.</p> |
| 5.0 Screen Index | <p><i>Property shown for TSL version 5.0.</i></p> <p>16-bit monitor screen index number.</p> <p>Hex \$FFFF is reserved. Set to 0 if unused.</p> |
| 5.0 Display Index | <p><i>Property shown for TSL version 5.0.</i></p> <p>16-bit UMD display index number</p> <p>Hex \$FFFF is reserved. Numbering starts from 0.</p> |
| Streaming: | Additional properties, <i>available if Streaming Video is selected.</i> |
| Source Path | <p>URL of the video stream from the multiviewer.</p> <p>Note: This Source Path property field is filled out automatically from details entered under Multiviewer > Properties in the Orbit Project screen.</p> |

Table 9 Video Auto-Tile Properties, in Wall Editor screen (Continued)

| Property | Description |
|-----------------------|--|
| GPI: | <p>Additional GPI properties. <i>Available if a tile's GPI property is selected.</i></p> <p>These tile properties simply bind tile properties to configured physical GPIO's. For example, this enables either:</p> <ul style="list-style-type: none"> • Status of a GPIO input (asserted/ not asserted) to be reflected on a tile. For example, a red lamp indicator when a GPIO input is asserted. <p>Or:</p> <ul style="list-style-type: none"> • A GPIO output to be driven from one or more tiles. For example, a GPIO output is asserted to alert user of a triggered alarm <p>Note: The multiviewer unit's physical GPIO's themselves must be independently configured via RollCall Control Panel (via the multiviewer's GPIO RollCall Template screen).</p> |
| GPI Mode | <p>Pull-down menu. Select:</p> <ul style="list-style-type: none"> • Input - Associate tile with a physical GPI input. Tile gets GPI input status (asserted/ not asserted) from the GPI input. • Output - Associate tile with a GPI output (GPO). Tile can assert a GPI output. <p>Note: A GPI output may be asserted by more than one Tile.</p> |
| GPI Port Index | <p>Select which target multiviewer physical GPIO to associate the tile with.</p> <p>Value is in range 1 to 4 for MV-8 series multiviewers.</p> |

Table 9 Video Auto-Tile Properties, in Wall Editor screen (Continued)



Note: The screenshots here show eight Audio Bars.

The number of audio bars shown depends on the values of the **Left Audio Bar Count** and **Right Audio Bar Count** properties.

Figure 58 Configure Left Audio Bars dialog (for eight audio bars) and tabs:

- a) PCM
- b) DolbyE
- c) Auto

2.11 Assign Inputs to a Wall

After a video **Wall** design has **Tiles** added, the next step is to connect each **Video Tile** to an individual multiviewer video input.

Consider the wall as one large display, rather than several individual monitor displays each connected to a multiviewer display output. View the entire video wall in the Orbit **Wall Editor** screen. (Use the 'Zoom Out (-)' and 'Scale to Fit' controls.)

Assigning inputs to tiles may be done automatically or manually.

2.11.1 Automatic Assignment of Inputs



This is carried out with the **Input Assignment** icon on the **Wall Editor** tool bar, See Figure 59. **Note:** The icon is grayed out when no tiles are selected.

The **Input Assignment** icon carries out a renumbering of all the video and TSL inputs on selected tiles, thus automatically assigning video inputs to selected tiles.

Note:

Re-numbering:

- Video Input re-numbering is carried out by Orbit on automatically-generated tiles, **Auto-Tiles**.
- Tile video input renumbering is not necessarily carried out on **Custom Tiles**.
To enable renumbering of the video input of a **Custom Tile**: Set the **Custom Tile's Configuration > Input** property to {Input} .
This uses Orbit's internal system variable "Input", which is automatically filled out (i.e. enumerated) when inputs are assigned.
- Re-numbering is applied to TSL property values which use the system variables "5.0 Display Index" or "3.1 Display Address".
- TSL screen index parameters (system variables "5.0 Screen Index") are set by Orbit.

To assign inputs:

1. Open a video wall in the **Wall Editor**.
2. Select all the **Tiles** to have inputs automatically assigned.

Note:

Selection/ Deselection of wall items:

The following 'select' / 'deselect' actions are used in the **Wall Editor**:

| Select / Deselect | Action |
|------------------------------|---|
| Deselect All | Click outside the schematic area. |
| Select | Click on an item to select it. |
| Add Select | Shift-Click on an item to add it to current selection. |
| Remove from Selection | Shift-Click on an <i>already-selected</i> item to remove it from the current selection. |
| Select All | 'Ctrl-A' selects all items. |
| Select an Area | Shift drag cursor on screen to describe a rectangular area and select all objects wholly or partly in the area. |

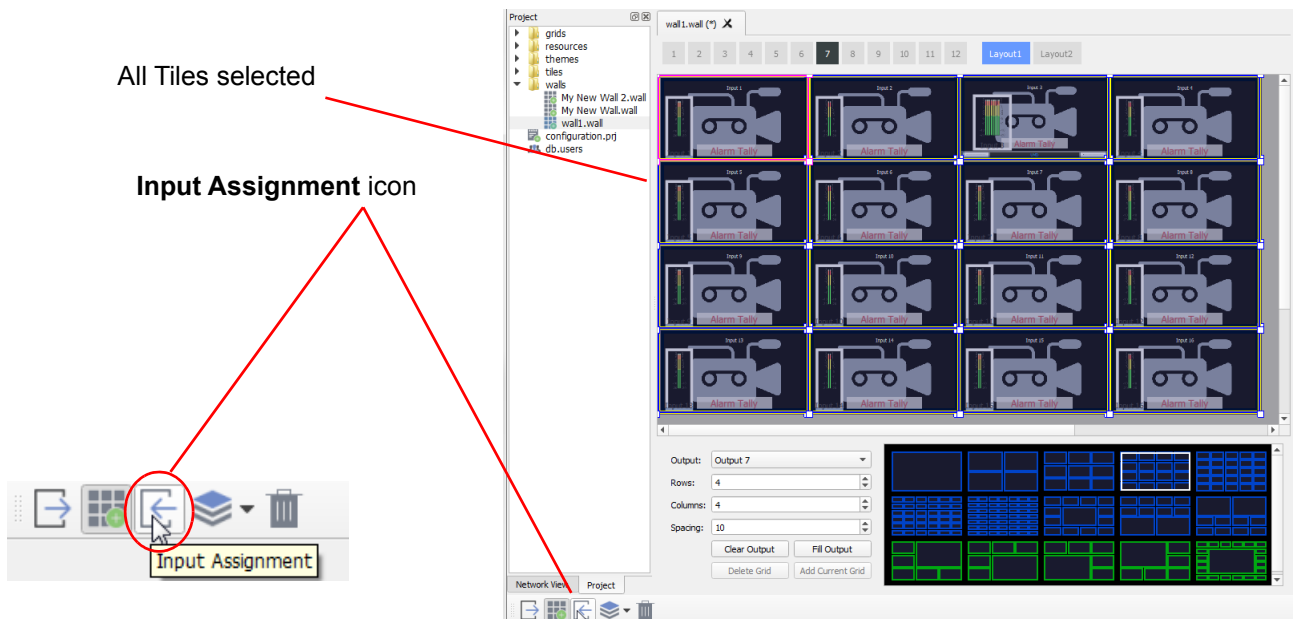


Figure 59 Input Assignment Icon (one or more tiles selected)

3. Click on the **Input Assignment** icon.
The **Input Assignment** dialog is displayed. See Figure 60.

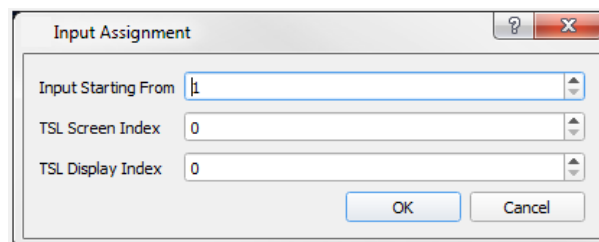
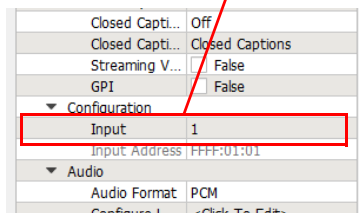


Figure 60 Input Assignment dialog

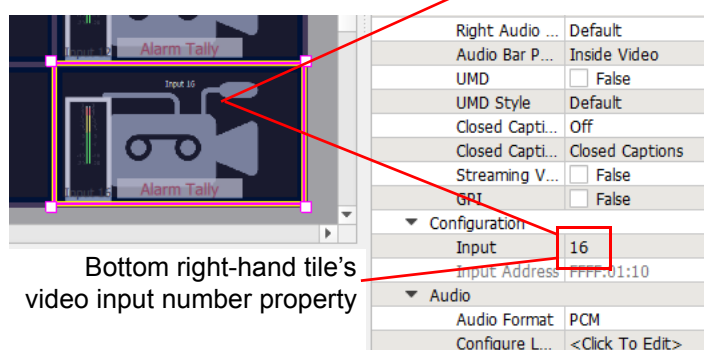
4. In the 'Input Starting From' text box,
enter the number of the Video Input to start numbering from.
(Video inputs will be re-numbered over the selected tiles, starting from this value and incrementing by one.)
 5. In the 'TSL Screen Index' text box,
enter the value to use for the TSL 5.0 screen index across UMD's in the selected tiles.
 6. In the 'TSL Display Index' text box,
enter the starting value for the TSL Display Index.
(TSL Display Index values will be re-numbered over the selected tiles, starting from this value and incrementing by one.)
- Note:** For TSL 3.1, use the 'TSL Display Index' value.
7. Click **OK**.
The video inputs and UMD TSL values assigned to the selected tiles are then re-numbered.

See Figure 61 for an example result.

Top left-hand tile's video input number property



Video Inputs assigned to Tiles, 1 to 16



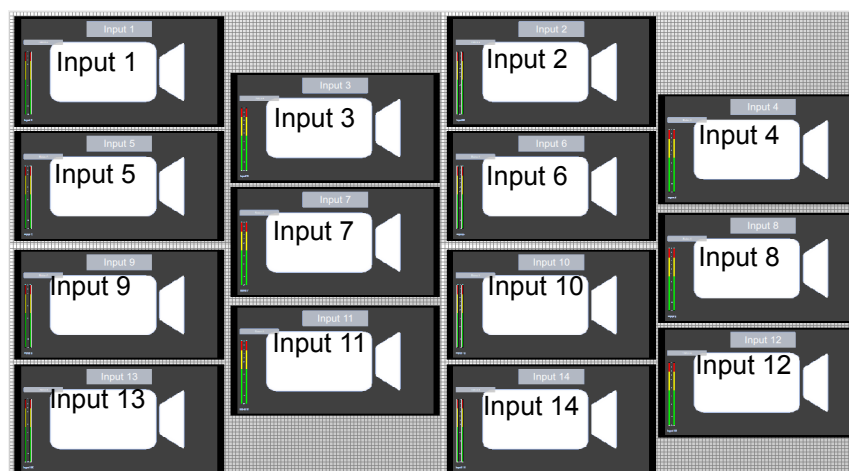
Bottom right-hand tile's video input number property

Figure 61 Video Inputs Assigned Automatically

Note:

Tile input re-numbering:

- Starts from the selected 'starting at' number.
- Re-numbers tiles in order from left-to-right, top-to-bottom. See Figure 62.



Note: Renumbering in order from left-to-right, top-to-bottom

Figure 62 Example Tile Video Inputs Renumbering Order

2.11.2 Manual assignment of video inputs

Tiles may also be manually renumbered.

1. Select a tile.
2. Modify the tile property **Tiles > Configuration > Input** property in the Properties Box.
3. Click **Save File** in the Orbit main tool bar to save any changes made.

2.12 Wall Layouts

More than one video wall layout may be designed for a single **Wall** design. All of a **Wall** design's layouts share the same assignment of multiviewer display outputs.

The **Wall Layouts** enable alternate arrangements of tiles to be used for the same wall design. See Figure 63.

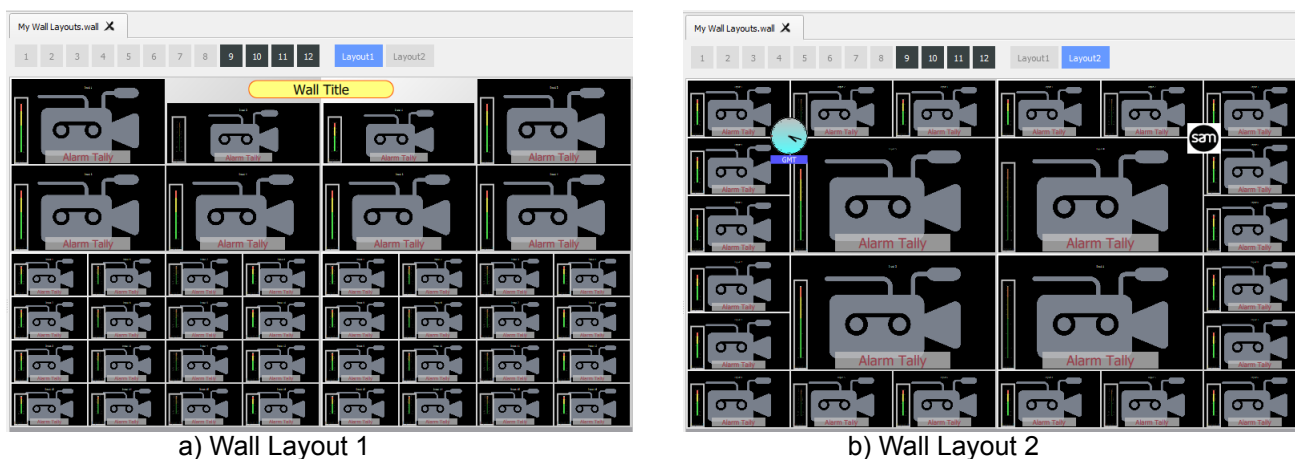


Figure 63 Two Wall Layouts for the same Wall

Up to 64 layouts are supported per wall.

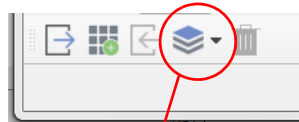
2.12.1 How to Create a new Wall Layout

To create a new layout in the **Wall Editor**:

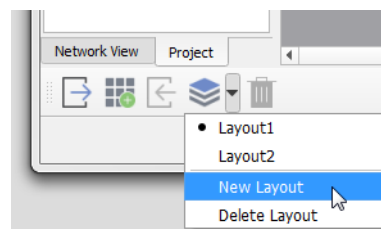
1. Start with an existing wall design open in the **Wall Editor**.
2. When changing layout views or creating new layouts, remember to save any changes in the current layout by clicking **Save File** in the Orbit main tool bar.

To create a new layout:

3. Click on the small triangle on the **Layout** icon, see Figure 64a. A pop-up menu appears. See Figure 64b.



a) Wall Layout Icon



b) Click Pop-up Menu

Figure 64 New Wall Layout:

a) Icon.

b) Pop up Menu.

- Click **New Layout** in the pop-up menu to create a new layout in the **Wall Editor**.

A new layout is shown, see Figure 65b, and a new “Layout 2” selection button.

Layout Selection Buttons

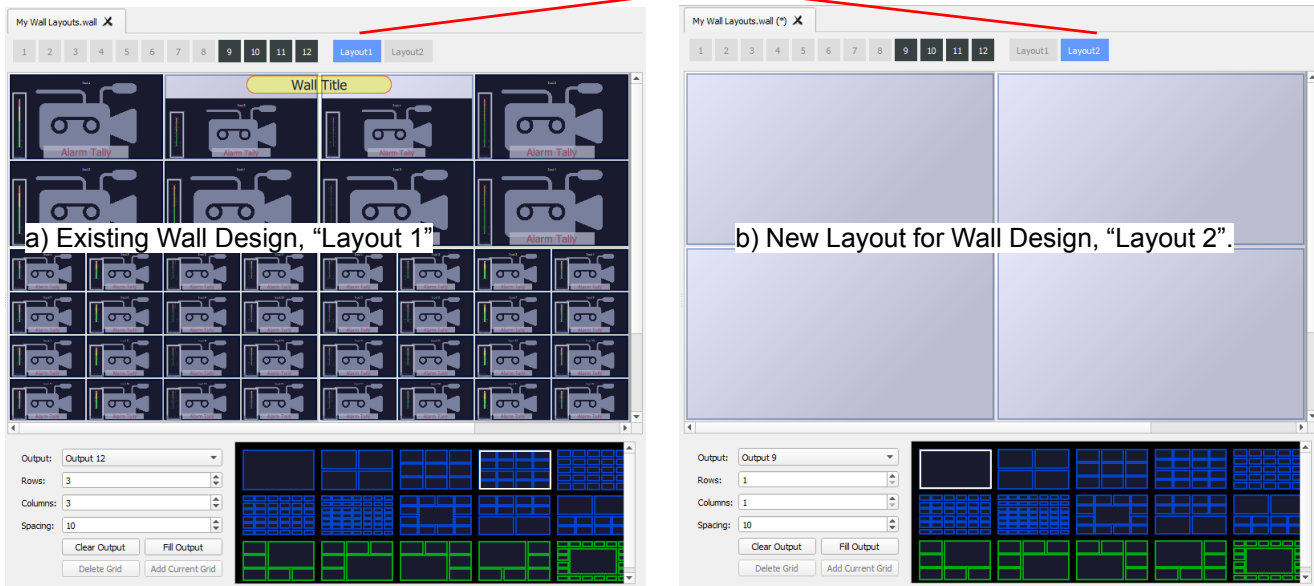


Figure 65 New Wall Layout:

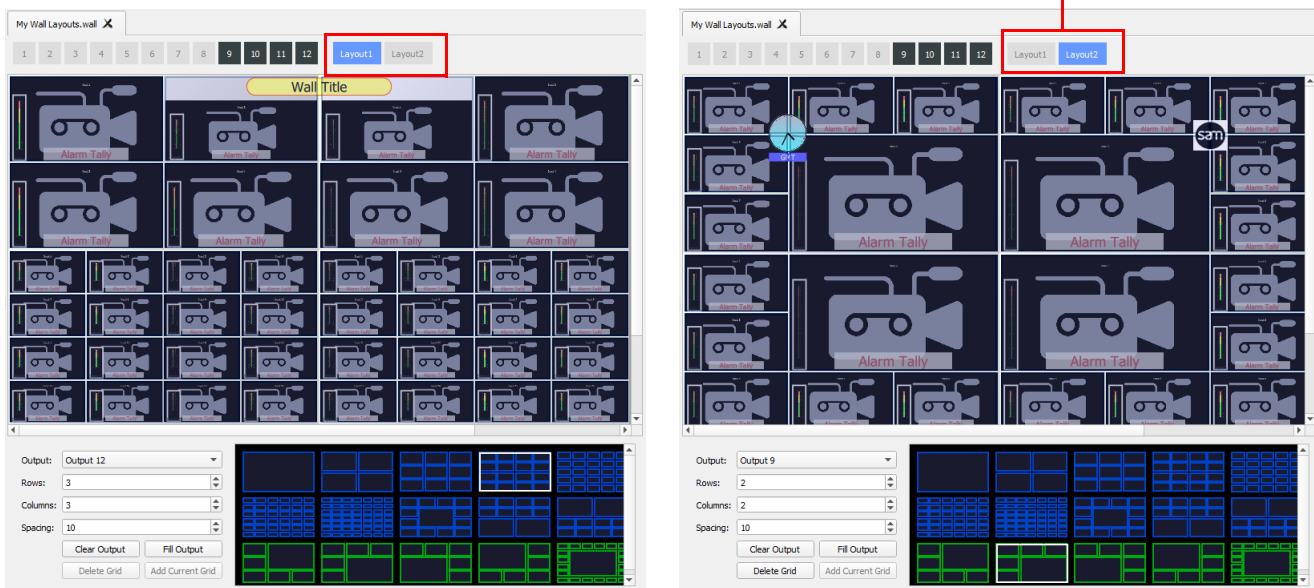
a) Existing Wall Design (Layout 1).

b) New Layout for Wall Design (Layout 2).

- Design the new wall layout just like designing a new wall: (Add **Tile Grids**. Then add **Tiles** to the grids, etc.)
- Click **Save File** in the main tool bar when all wall changes are done.

There are now two wall layouts associated with the wall design, “Layout 1” and “Layout 2”. See Figure 66.

Layout Selection Buttons



a) Existing Wall Design, “Layout 1”

b) New Layout for Wall Design, “Layout 2”.

Figure 66 New Wall Layouts:

a) Existing Wall Design.

b) Second Layout for Wall Design.

2.12.2 Switch Wall Layouts in the Orbit Wall Editor

The wall **Layouts** exist in parallel in an Orbit project. Any layout for a wall may be edited in the **Wall Editor** but only one can be edited at any one time.

To switch layouts in the **Wall Editor**:

- Click the **Layout Selection** buttons at the top of the screen to switch between wall layouts in the **Wall Editor**. See Figure 67.



Figure 67 Layout Selection Buttons

Or

- Click on the **Layout** icon in the **Wall Editor Tool Bar** and use the pop-up Menu to select a wall layout. Click on the layout name required. See Figure 68.

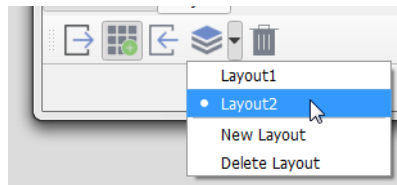


Figure 68 Layout Selection Menu

Or

- Right-click in the **Wall Editor** schematic area, but *outside of the wall design*. See Figure 69. A pop-up menu appears.

Click **Switch Layout** to access a pop-up Layout Selection Menu, see Figure 69.

Right-click *outside of the wall design*, but within the schematic area.

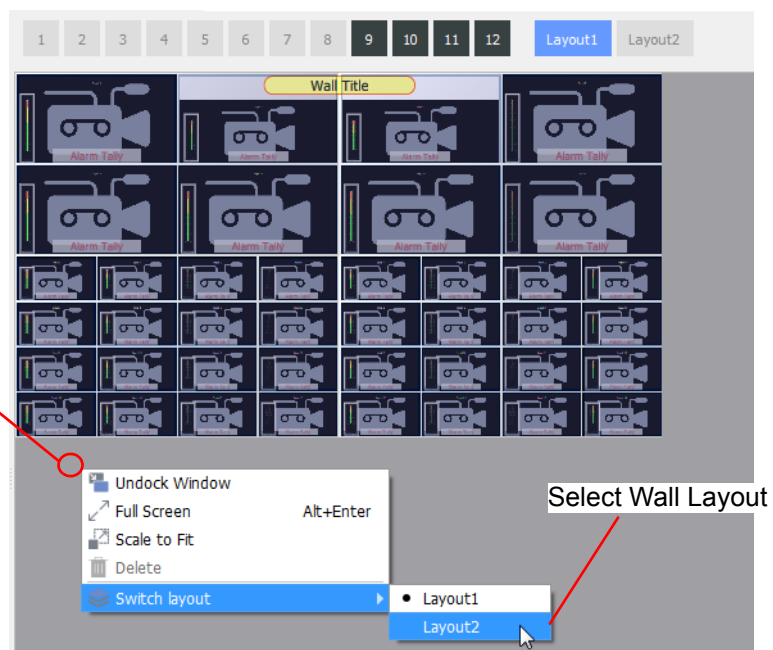


Figure 69 Switch Wall Layout using a Right-Click menu

2.12.3 Switch Wall Layouts in Orbit in Run Mode (MV-800-DT License)

When viewing a live video wall in Orbit, right-click on the wall to change wall layout.

Right-click on **live Wall**. A drop-down menu appears.

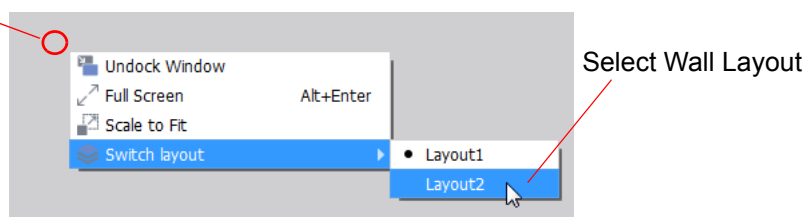


Figure 70 Live Video Wall in Orbit, Changing Wall Layout

2.12.4 Switch Wall Layouts on a Multiviewer Unit

An Orbit project may contain two or more different layouts for a wall. When a project is deployed on a multiviewer, the wall layout to be used can be selected remotely. Only one wall layout may be selected for use on a multiviewer at any one time.

Wall Layouts can be changed from the Multiviewer's RollCall Control Panel template via the Layout template screen.

Refer to the appropriate Multiviewer User Manual, RollCall Configuration section for full information.

2.12.4.1 Multiviewer RollCall Layout Screen Overview

A RollCall Layout screen allows the user to select which video wall layout to apply to a multiviewer display output.

In a Orbit multiviewer project:

- Each video **Wall** has a **Name** property.
(Set to "Wall 1", "Wall 2" etc. by default in Orbit.)
The **Wall Name** may be edited in Orbit by the user.
- Each **Wall Layout** has a **Name** property. This is not editable by the user.
This is set to "Layout1", "Layout2" etc. in Orbit

Note:

These names are case-sensitive.



Figure 71 MV-8 Series Multiviewer's Typical RollCall Layout Screen

| Setting | Description |
|---------------------------|--|
| Wall 1 to Wall 12: | |
| Layout | Text box. Enter the wall layout name to use. (Case sensitive) |
| Layout Transition: | |
| Fade Through Black | Check box. Controls the type of on-screen transition seen when changing between wall layouts. Selected: Fade down to black; then fade up to new wall layout. Unselected: Cut from current to new wall layout. |

Table 10 TSL Mode Settings

Remember to press the **S** button to save a change.

Note:

“S” and “P” buttons - After entering information in each text box, always click on the adjacent “S” button or press “return” to locally save the new setting. Do this for each text box. (Note: Clicking on the “P” button will return the setting to its preset default value).

3

3 Tile Editor - Tiles and Widgets

Tile Editor - Tiles and Widgets

| | |
|---|-----|
| Introduction | 74 |
| Terminology | 75 |
| Opening a Tile Design in the Tile Editor | 76 |
| Creating a New Tile | 76 |
| Creating a New Tile Design from an Auto-Tile | 80 |
| Opening an Existing Tile Design | 81 |
| Opening Tiles using the Project View | 82 |
| Tile Editor Screen | 84 |
| Tile Editor Tab | 85 |
| Main Tool Bar items | 86 |
| Properties Box | 86 |
| Selecting Graphical Objects when Editing in Tile Editor | 87 |
| Placing Widgets | 88 |
| Widgets Tool Bar | 88 |
| Add a Widget to a Tile or Schematic | 88 |
| Reposition/Resize Widgets | 89 |
| Schematic 'Right-Click' Context Menus | 92 |
| Tile Editor - Advanced Tool Bar | 99 |
| Auto Fill Facility | 101 |
| Example - Labeling Widgets in a Numbered Sequence. | 102 |
| Auto-Tiles for Multiviewers | 105 |
| Video Auto-Tiles - 3G/HD and 4K | 107 |
| Video Auto-Tile - Style Properties | 108 |
| Video Auto-Tile - Properties | 109 |
| Video Auto-Tile - Appearances | 110 |
| Clock Auto-Tiles | 111 |

3.1 Introduction

Orbit offers some built-in tile designs. These are automatically-generated (**Auto-Tiles**) in the case of multiviewer video walls. The **Auto-tiles** cover the majority of tile use-cases and they have built-in, configurable functionality.

For specific use-cases, a tile's *appearance* may require customizing. This can be accomplished with Orbit's **Themes** and widget **Styles**. (See "Theme Editor - Widget Styles" on page 112.)

For other use-cases, a tile's *functionality* may require customizing or a new type of tile may be required. This can be done with a **Custom Tile**.

For a multiviewer unit, the video wall comprises various tiles which may be configured in the Orbit **Wall Editor** or edited in the Orbit **Tile Editor**:

- **Tile Changes in the Wall Editor:**

Tile edits done in the **Wall Editor** are restricted to changes to some configuration tile properties.

The changes made affect just the tile being edited in the wall, 'wall-level' changes: They do not affect every tile of the same type.

- **Tile Edits in the Tile Editor:**

Tile edits done in the **Tile Editor** may change all of a tile's properties and the properties of widgets that it may contain, including sizes and positions.

The changes made are at 'tile-level' and affect all tiles of the same type.

The **Tile Editor** is used to design and configure a **Custom Tile** (for example, for a customized multiviewer video wall). Graphical widgets are assembled on a tile and configured.

Note:

Creating Custom Tiles:

It is recommended that a **Custom Tile** is created using an **Auto-Tile** as a starting point. This is because **Auto-Tiles** already have some built-in functionality set up ("under the hood"). For example, functionality for the audio bars and UMD's widget.

3.2 Terminology

| Term | Definition |
|----------------------------|--|
| Tile | <p>A rectangular area on a screen, comprising a set of graphical widgets. Tiles are used on multiviewer video walls.</p> <p>On a video wall, the tile usually displays video and other supporting information. For example:</p> <ul style="list-style-type: none"> • Video picture with audio level bars. • Time, images, labels or text. |
| Schematic | Similar to a Tile , but for non-multiviewer applications (MapView). |
| Widget | <p>On-screen graphical element used to display information.</p> <p>Widgets include: Audio bars, Clocks, Tally LEDs, UMD's and Video sources.</p> |
| Wall Editor | <p>A schematic editor screen allowing Tiles to be added, moved and deleted on a video wall.</p> <p>Some Tile properties may be changed in the Wall Editor; changes apply to the particular Tiles being edited.</p> |
| Tile Editor | <p>A schematic editor screen allowing graphical widgets to be added, moved and deleted on a Tile.</p> <p>Changes apply to all Tiles of that type on the video wall(s).</p> <p>Tile edits via the Tile Editor result in a Custom Tile.</p> |
| Custom Tile | <p>A Tile which is generated by the user in the Tile Editor. The Tile contains a user-defined set of graphical widgets, each with adjustable properties.</p> <p>Editable in the Tile Editor.</p> |
| Auto-Tile | <p>A video wall tile which is generated automatically by the Orbit software, containing a fixed set of graphical widgets with built-in functionality. For example, a video Auto-Tile comprises video source, UMD and audio bars widgets.</p> <p>Select an Auto-Tile in a wall with a single-click to see its properties in the Properties box.</p> <p>Note: Double-clicking on an Auto-Tile in a wall, opens the Tile in the Tile Editor. Changes to Tile and widget size, position and properties. can be made. However, a Custom Tile results from this editing.</p> |
| Properties | <p>Attributes of a graphical widget.</p> <p>For example, background color, border width.</p> |
| Extended Properties | Further widget properties. |
| Variables | <p>Project-wide or Tile-wide values held by the Orbit project.</p> <p>User-defined variables can be created.</p> <p>A variable's value may be used in property values. For example, {Input}.</p> |

Table 11 Terminology

3.3 Opening a Tile Design in the Tile Editor

Tile editing is done in the **Tile Editor**, accessed from the Orbit **Project Home** screen.

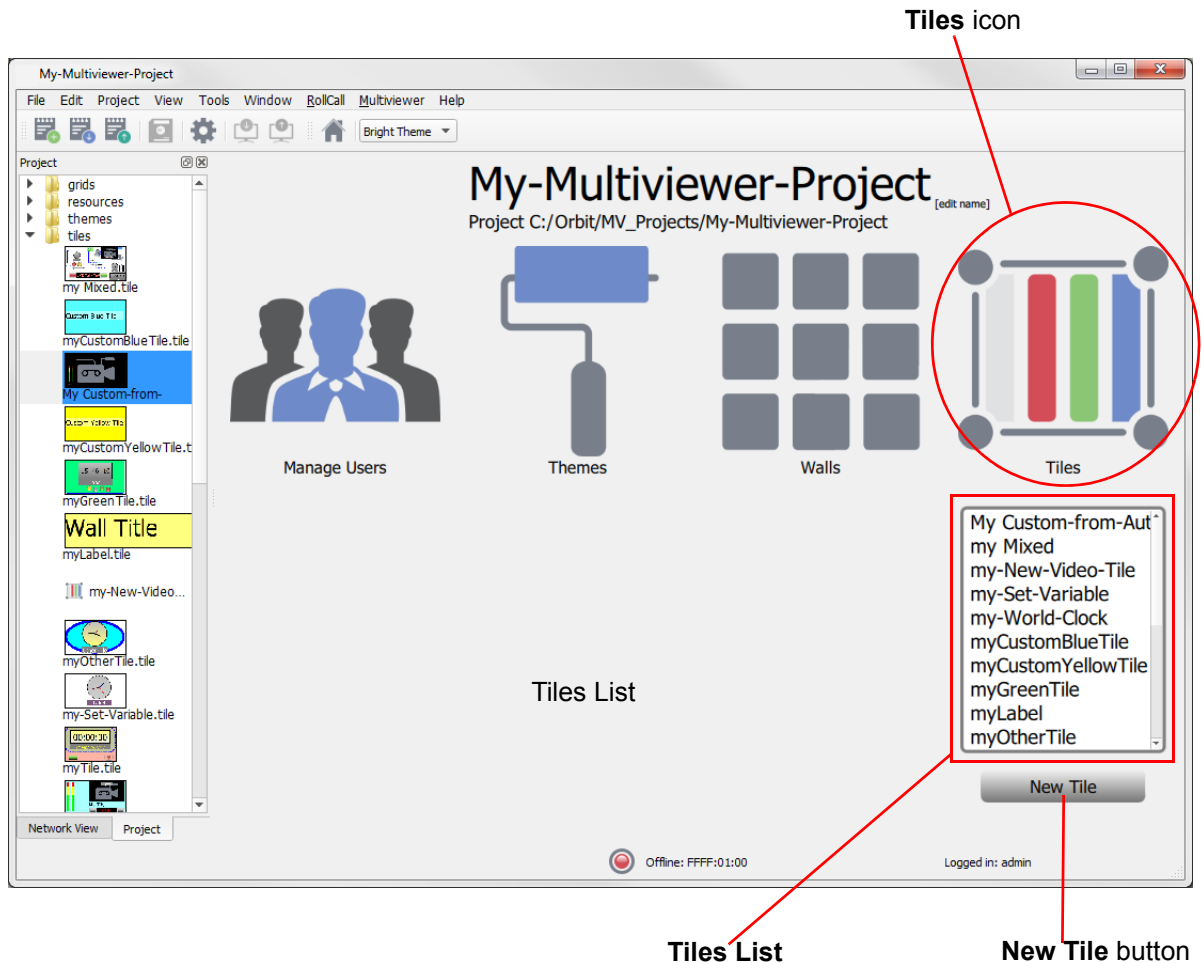


Figure 72 Orbit Project Screen with Tiles Icon

3.3.1 Creating a New Tile

Note:

Creating Custom Tiles:

It is recommended that a **Custom Tile** is created using an **Auto-Tile** as a starting point (rather than creating a **Custom Tile** from an empty tile).

This is because **Auto-Tiles** already have some built-in functionality set up ("under the hood"), for example, to give the audio bars and UMD's widget their functionality.

To create a new tile design:

1. Either:

- In a Orbit **Project Home** screen, click the **Tiles** icon. Click the **New Tile** button, located below the **Tiles List**.

Or:

- In the **Project View**, right-click on the 'tiles' sub-folder item. Click **Add Tile**.

A **New Tile** dialog appears. See Figure 73.

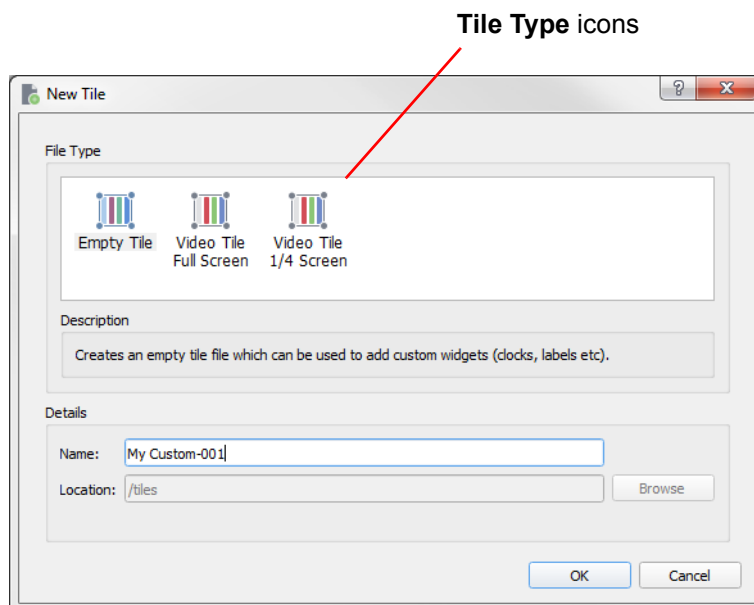


Figure 73 New Tile Dialog

- Click on the **Tile Type** icon to select the type of new **Tile** required, see Table 12.

| Tile Type | Description |
|----------------------------------|--|
| Empty tile | A blank empty tile. Tile can be used to add other custom widgets (Clock, Labels etc.) Size: 960 x 540 pixels. |
| Video Tile Full Screen | A generic video tile, HD screen sized. A Tile with a single video widget. Size: 1920 x 1080 pixels. (See Figure 74.) |
| Video Tile Quarter screen | A generic video tile. A Tile with a single video widget. Quarter-HD-screen size: 960 x 540 pixels. |

Table 12 New Tile Types

- Enter a short descriptive name for the new tile design in the **Name** text box.
- Use the location shown in the **Location** text box.

Note:

Location:

The location of the new tile design file will default to the '**tiles**' sub-folder in the Orbit project, shown in the **Project View**.
The path to this file location is shown in the **Location** text box: Use this location.

- Click **OK** to open the new tile in the **Tile Editor** screen, the new tile is opened in the **Tile Editor** screen. See Figure 74a, b or c.
(Or click **Cancel** to abort and return to the Orbit **Project Home** screen.)

Note:

Tile file:

All information relating to a tile will be stored in a file within the Orbit project. The file has a '.tile' extension. A project may contain more than one tile design.

The tile files are shown in the **Project View**, when expanded.

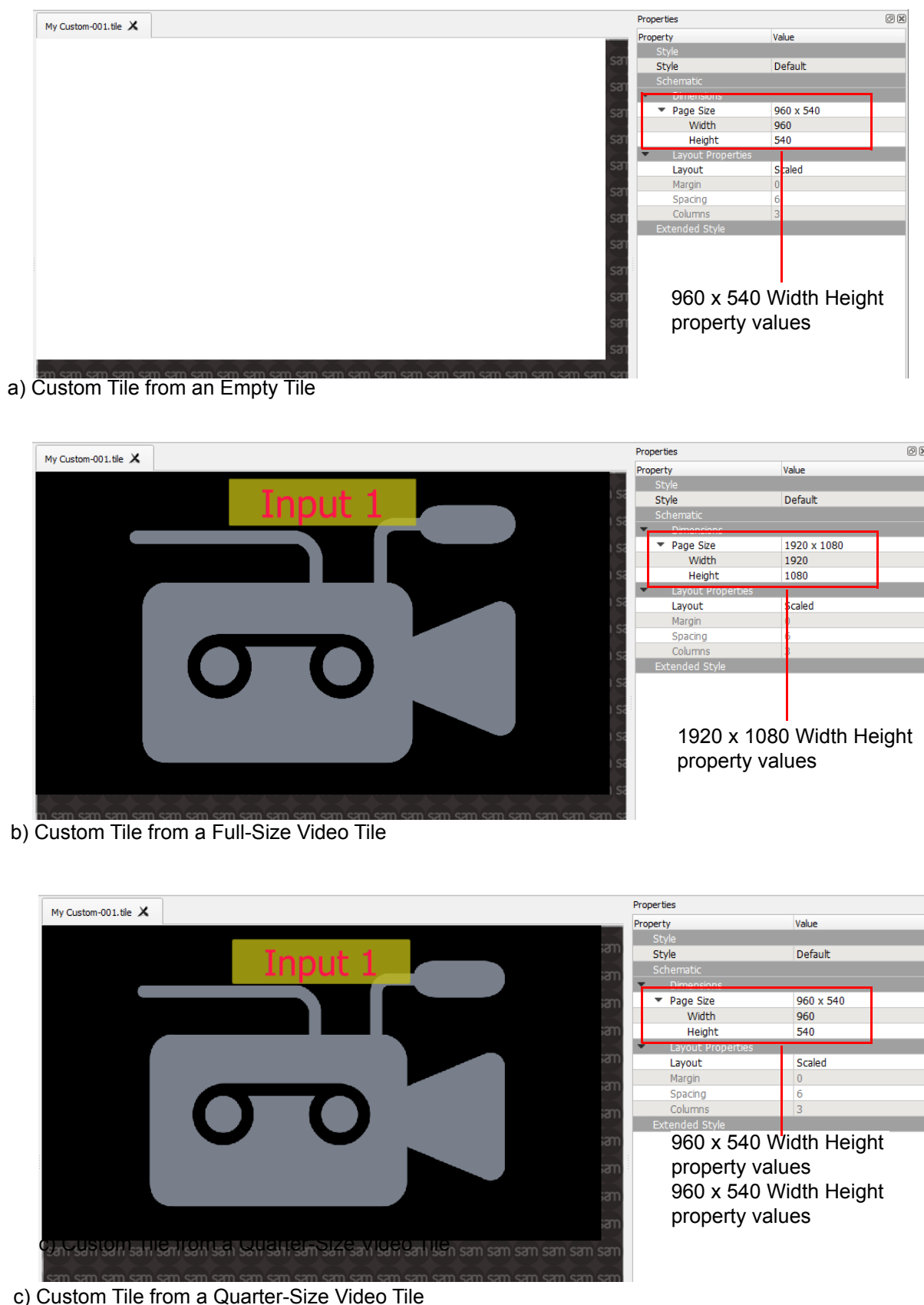


Figure 74 Example New Tiles:

- a) Empty Tile.
- b) Full-Size Video Tile, 1920 x 1080 pixels.
- c) Quarter-Size Video Tile, ie Full Screen, 960 x 540 pixels.

6. The **Tile Editor** screen is displayed, see Figure 75.

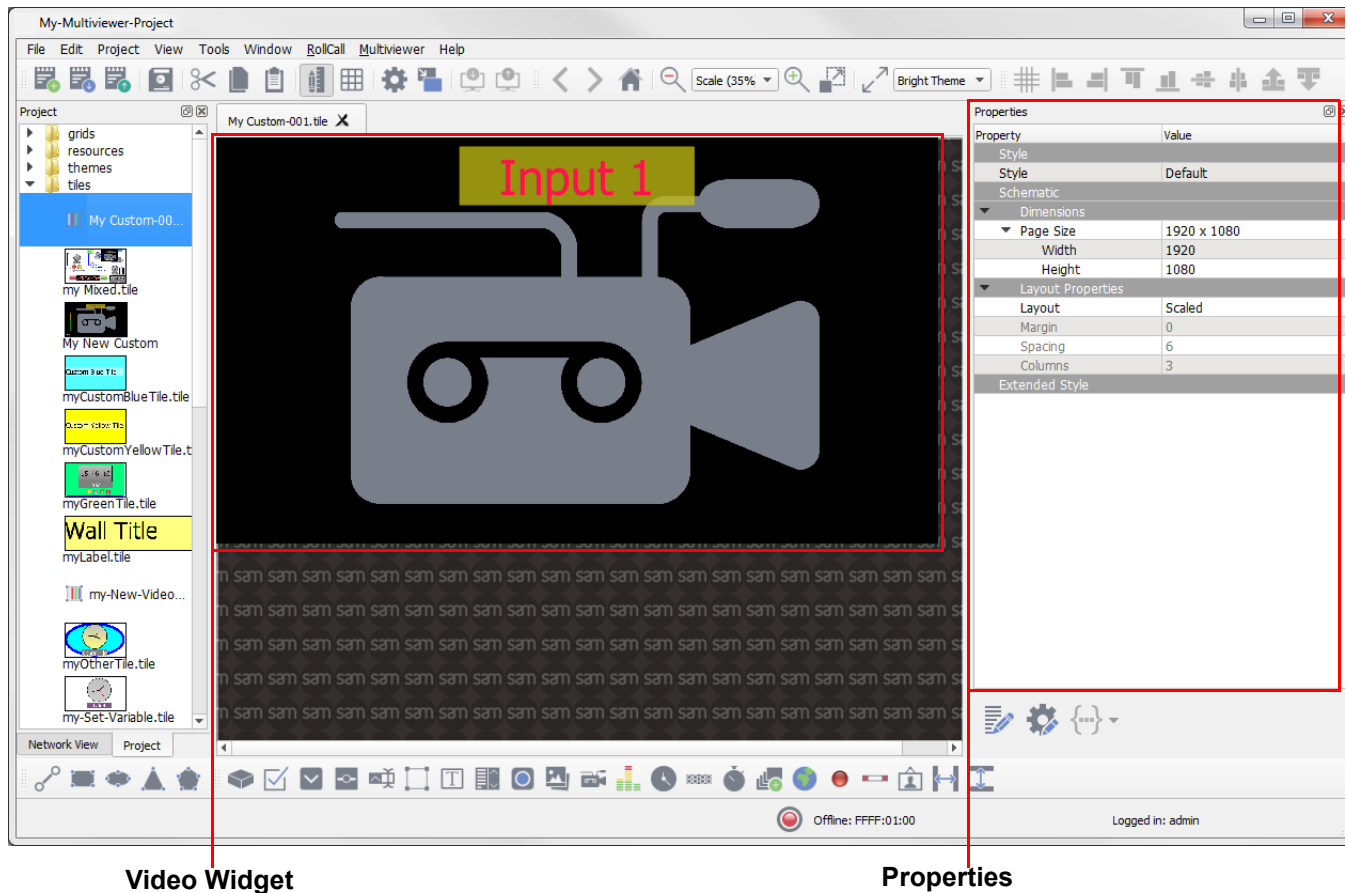


Figure 75 Tile Editor screen - for a New Tile Video Tile

A new Tile design has been created.

The **Tile Editor** screen is described in Section 3.4 “Tile Editor Screen” on page 84.

Note:

Widget properties:

Click on a widget to select it and see its properties.

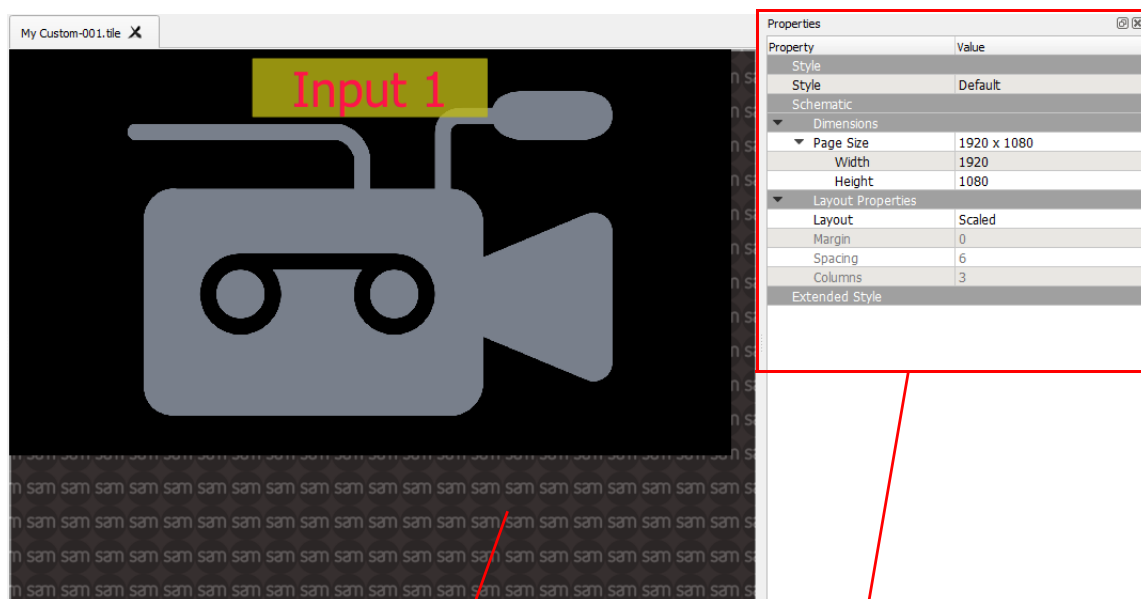
Tile properties:

To see the properties of the **Tile** itself, rather than properties of any widgets within it, click in the outer **Tile Editor** schematic background area. See Figure 76a.

(This deselects all graphical objects, for example, tiles or widgets. Then the properties of the tile schematic are shown in the properties box.)

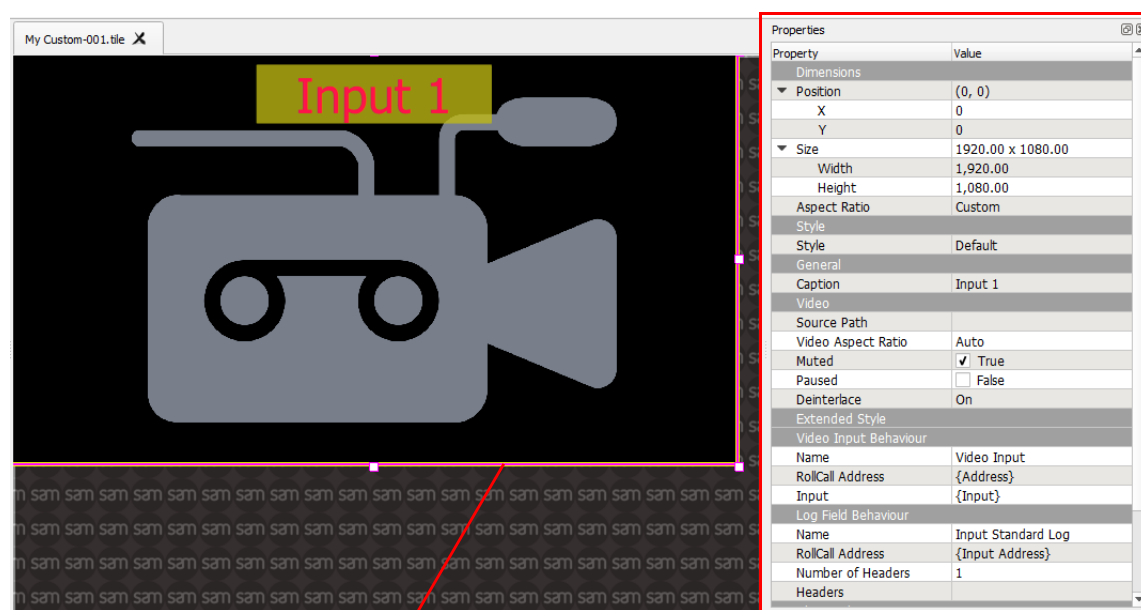
The **Tile** in Figure 75 contains a rectangular video widget.

7. Click on a widget to select it and see its widget properties. See Figure 76b



a) Tile Properties. (Tile Editor Background Selected)

Tile Properties shown



b) Widget Properties. (Widget Selected)

Widget Properties shown

Figure 76 Tile Editor screen:**a) Tile Properties.****b) Widget Properties.**

3.3.2 Creating a New Tile Design from an Auto-Tile

To create a new tile from an **Auto-Tile**:

1. Open a wall design which is populated with video **Auto-Tiles**.

2. Double-Click on an **Auto-Tile** in the wall.
A **Create Custom Tile** dialog appears, see Figure 77.

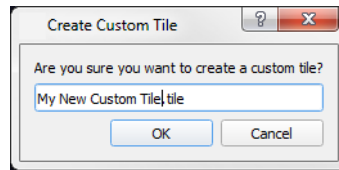


Figure 77 Create Custom Tile Dialog

3. Enter a name for the new **Custom Tile**.
4. Click **OK** if you are sure you wish to create a **Custom Tile** from this **Auto-Tile** at this wall location. (Click **Cancel** to cancel custom tile creation.)

A **Custom Tile** is created and the **Tile Editor** opens. See Figure 78.

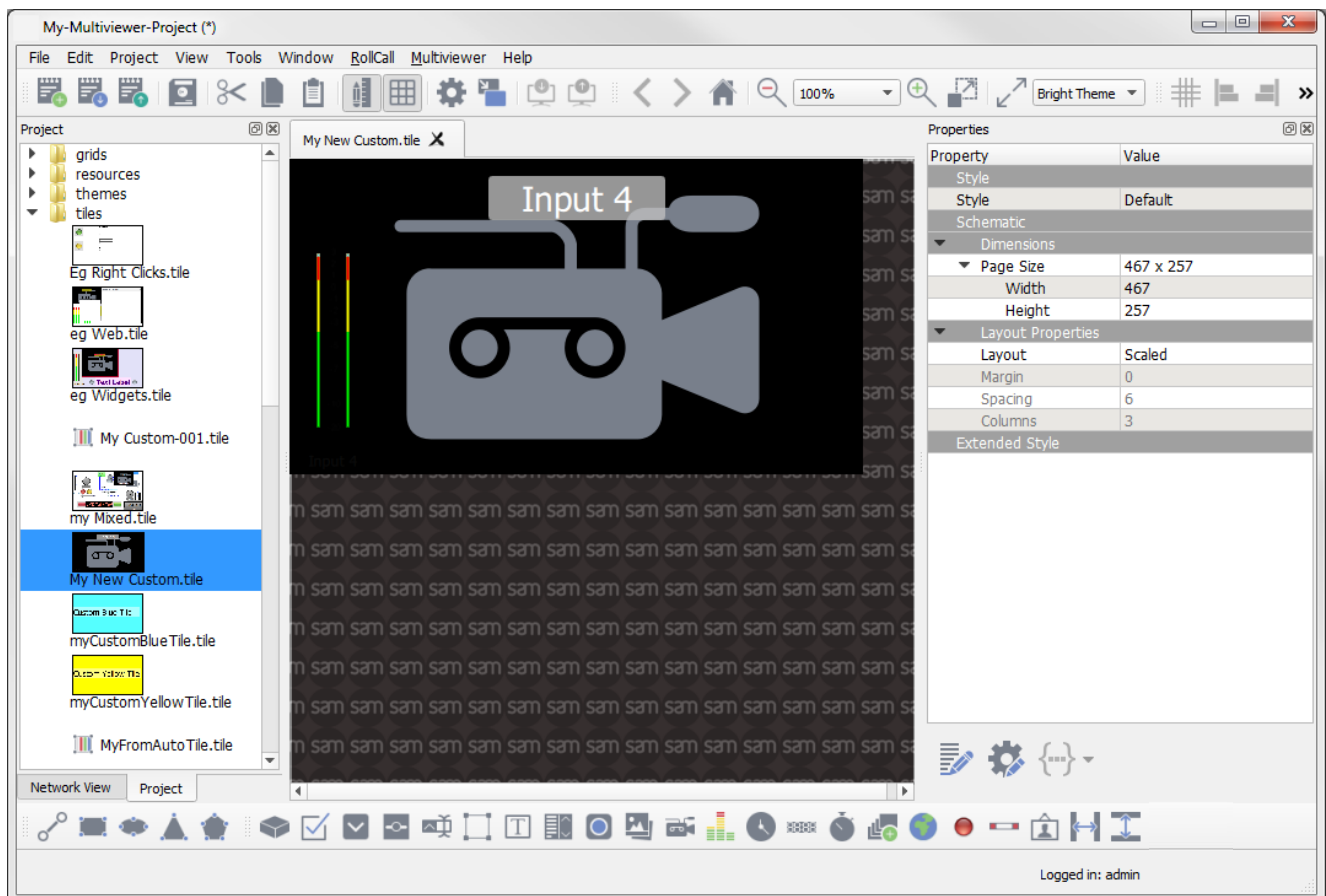


Figure 78 Tile Editor screen - Custom Tile created from a Video Auto-Tile

3.3.3 Opening an Existing Tile Design

1. In a Orbit **Project Home** screen, click the **Tiles** icon. See Figure 72.
A list of **Tile** designs appears below the icon.
2. In the **Tiles List**, double-click on the **Tile** design name to be opened.
The **Tile Editor** screen is displayed with the **Tile** design showing.

The existing Tile design has been opened.

The **Tile Editor** screen is described in Section 3.4 “Tile Editor Screen” on page 84.

3.3.4 Opening Tiles using the Project View

To create a new tile using the **Project View**:

1. In the **Project View**, right-click on the ‘**tiles**’ sub-folder item.
2. Click **New Tile** in the pop up menu. See Figure 79

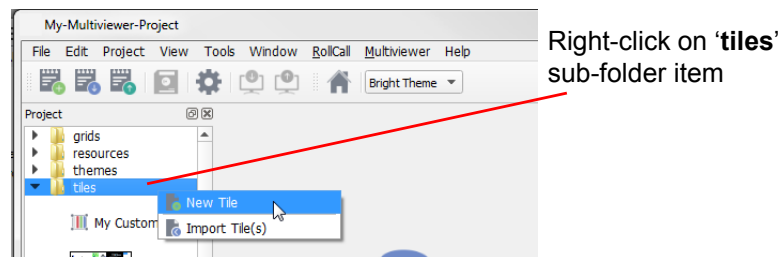


Figure 79 Project View - New Tile

3. A **New Tile** dialog appears. (See Figure 73 on page 77.)
4. Select the **Tile** type required.
Enter a **Name** for the new tile.
Click **OK**.

The new **Tile** is opened in the **Tile Editor** screen

To open an existing tile using the **Project View**:

1. Expand the **Project View** ‘**tiles**’ sub-folder tree view.
2. Right-click on a **Tile** name.
A drop-down menu appears. See Figure 80.

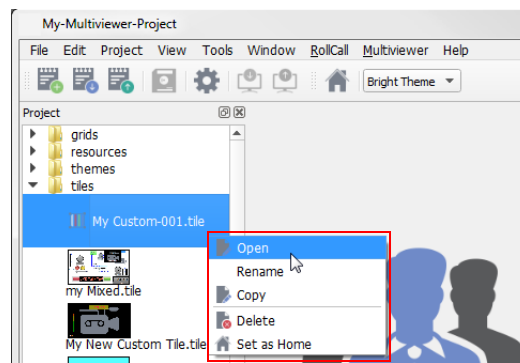


Figure 80 Project View - Open a Tile

3. Click **Open** in the pop up menu.
The **Tile** design is opened in a **Tile Editor** screen.

The **Tile Editor** screen is described in Section 3.4 “Tile Editor Screen” on page 84.

Further Project View operations:

Additionally, from the **Project View**, **Tiles** can be:

- Imported from another Orbit project.

- Renamed.
- Copied and Pasted.
- Deleted.
- Set as the Home Schematic.

(See the 'Orbit - Introduction User Manual' for information about **Project View** menus for these functions.)

3.4 Tile Editor Screen

The **Tile Editor** allows a user to view and edit **Tiles**. The **Tile Editor** screen is accessed from the Orbit **Project Home** screen. Figure 81 shows the **Tile Editor** screen and an opened **Tile**.

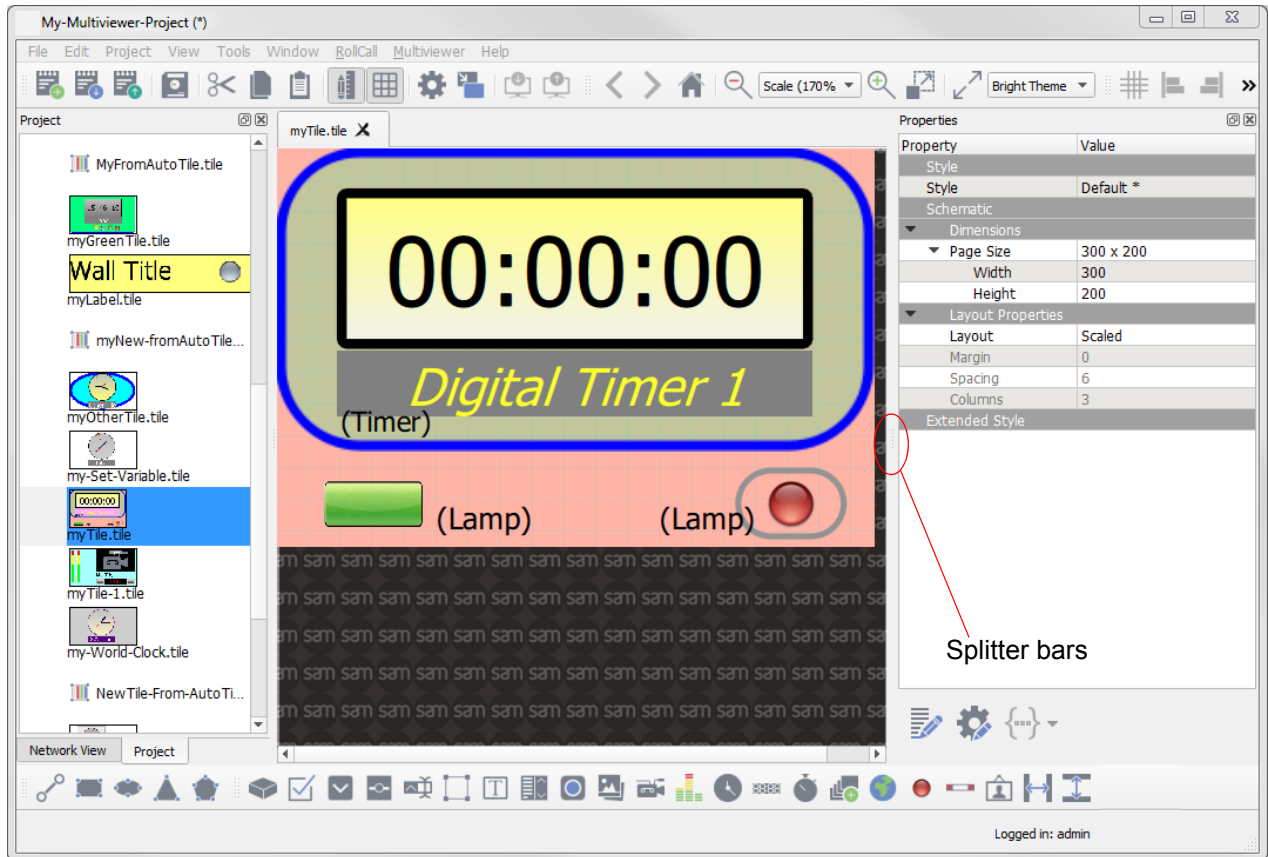



Figure 81 Tile Editor screen

To adjust the relative sizes of the **Tile Editor** schematic area and **Properties** box:

1. Position the cursor between the areas of the screen, on the splitter bars.
2. When the cursor changes shape to , drag the cursor to resize the areas.

The **Tile Editor** screen uses several tool bar icons in the Orbit main tool bar and has other tool bars lower down the screen.

Figure 82 highlights some of the features of a **Tile Editor** screen.

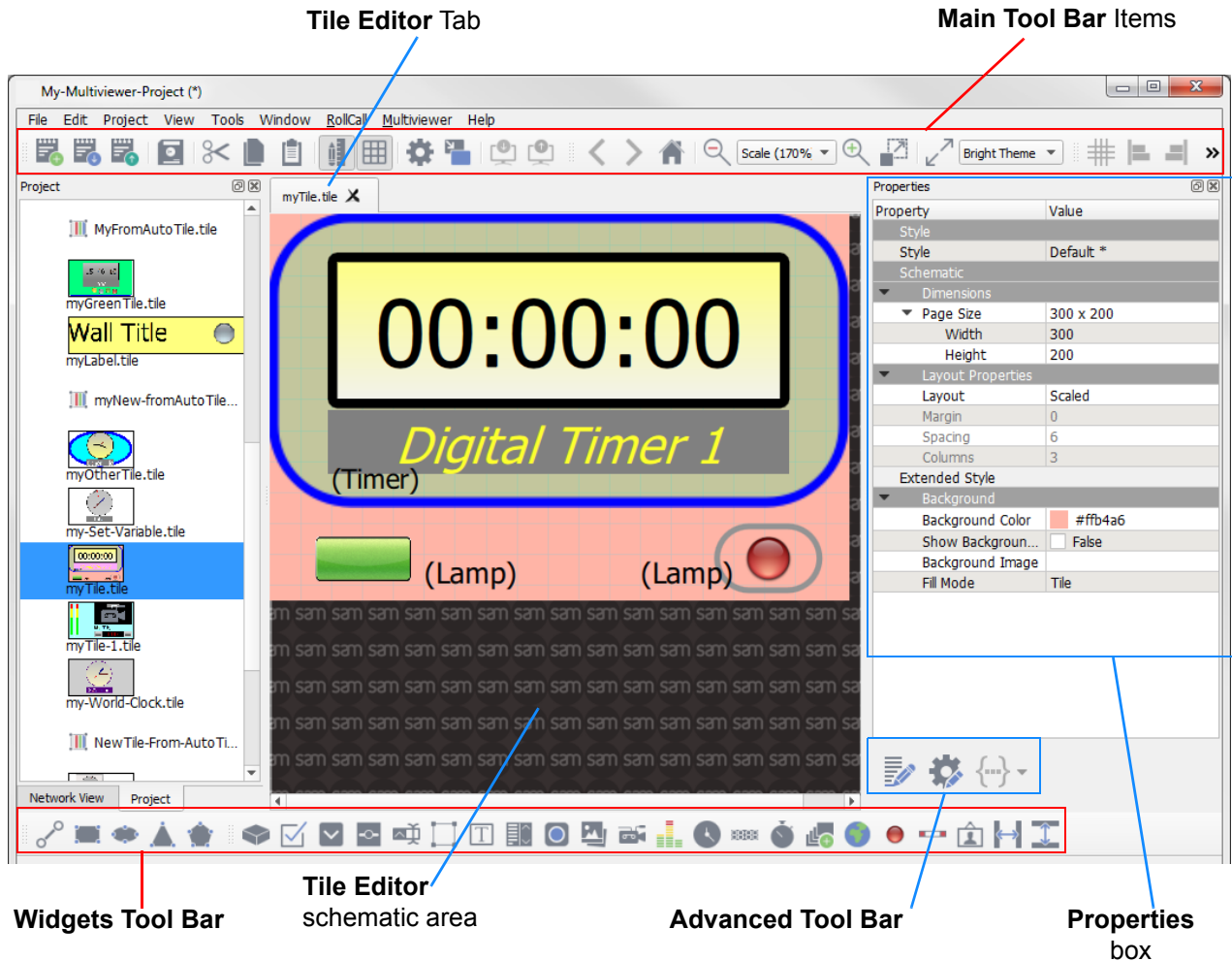


Figure 82 Tile Editor screen - annotated

3.4.1 Tile Editor Tab

The **Tile Editor** screen is identified by a tab; the tab arrangement allows several **Tile Editor** screens to be open at once. Clicking on the tabs moves between the **Tile Editor** screens.

Each tab is labeled with the **Tile** design name and a ".tile" suffix. A file with this name appears in the Orbit project and is shown in the **Project View**, in the 'tiles' sub-folder.

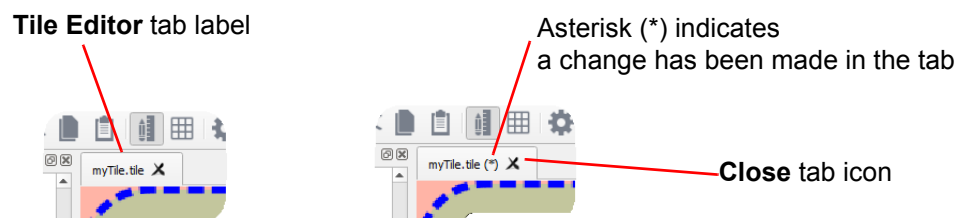


Figure 83 Tile Editor Tab - Label, (*) and Close Icon

A **Tile Editor** tab may be closed by clicking on the **Close** tab icon. If changes have been made in the tab, a prompt is shown, asking to save any changes made.

3.4.2 Main Tool Bar items



All the **Main Tool Bar** items are described in the 'Orbit - Introduction User Manual'.

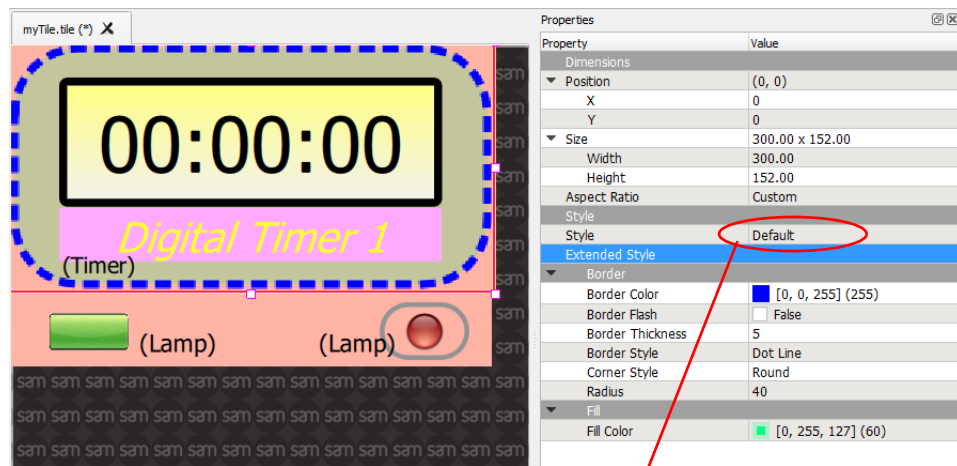
Many icons are generally applicable to editing graphical on-screen items like **Tiles**, schematics and widgets.

3.4.3 Properties Box

Properties of a selected **Tile** item, usually a widget, are shown in the properties box. If no widgets are selected, then the properties of the **Tile** itself are shown.

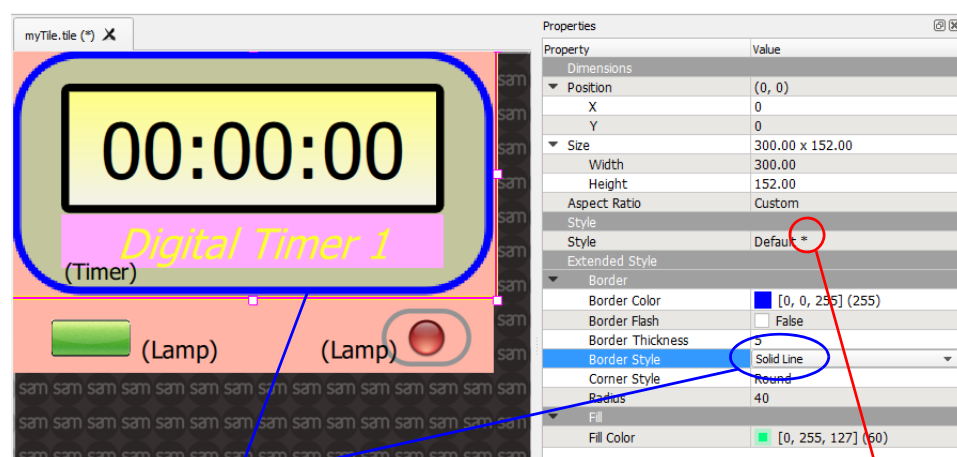
Property values may be changed. An asterisk beside a property drop-down value indicates a style change has been made locally to a property. See Figure 84 for a style property example. Such a local change will affect only the tile being edited. And all instances of the tile on a wall will be affected.

Expand or collapse items in the **Properties Box** using the in-line triangular expand/collapse icons ( ).



a) Style Property unchanged.

Style property value *without* an asterisk, indicating no local style changes have been made



Change made to **Border > Border Style** property (changed to a solid line)

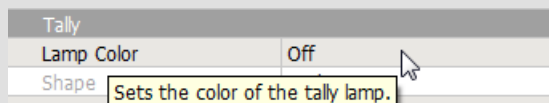
Asterisk * indicates style property for tile has been locally changed.

b) Style Property changed ('Border Style' has changed).

Figure 84 Change in a Style Property *

Note:**Tool Tips:**

Hover the cursor over a **Property value** to see a Tool Tip.



3.4.4 Selecting Graphical Objects when Editing in Tile Editor

Note:**Selection/ Deselection of graphical items when editing items in the Tile Editor:**

| Select / Deselect | Action |
|------------------------------|--|
| Deselect All | <p>Click outside the schematic area.</p> <p>Or press the 'Esc' keyboard key.</p> |
| Select | <p>Click on an item to select it.</p> <p>Alt-Click selects another item:</p> <ul style="list-style-type: none"> • In a stack of items. • Or selects the Group Layout that an item belongs to. |
| Add Select | <p>Shift-Click on an item to add it to current selection.</p> <p>Shift-Alt-Click to add another item from a stack of items to the current selection.</p> |
| Remove from Selection | Shift-Click on an <i>already-selected</i> item to remove it from the current selection. |
| Select All | 'Ctrl-A' selects all items. |
| Select an Area | Shift drag cursor on screen to describe a rectangular area. All objects wholly or partly in the area will be selected. |

3.5 Placing Widgets

3.5.1 Widgets Tool Bar

The **Tile Editor** allows various graphical widget objects to be dragged onto the tile being designed. All the available widgets and graphical objects are on the **Widgets Tool Bar**, see Figure 85.

Widgets are used on Orbit **Tiles** and on Orbit **Schematics**.

Note:

The Orbit application may be used to define graphical control surfaces in addition to designing multiviewer video wall layouts. Some of the available widgets apply more to these control surface applications than to traditional multiviewer video wall designing.

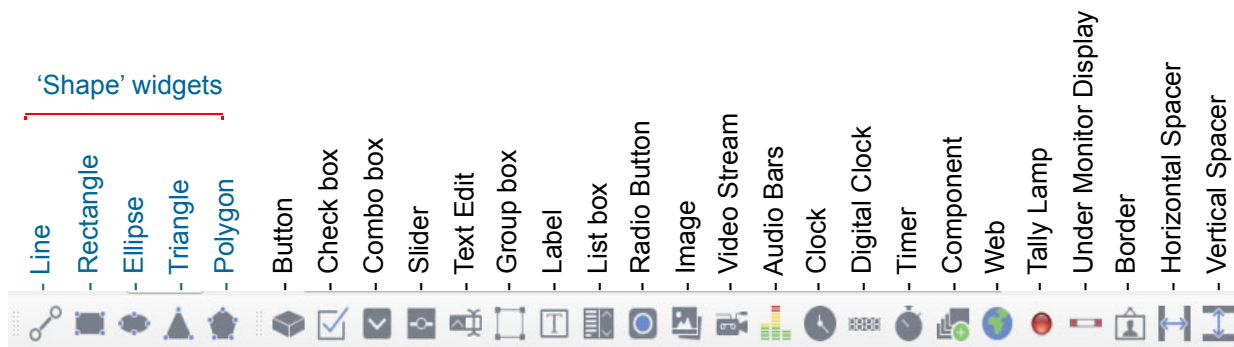


Figure 85 Widgets Tool Bar, annotated icons

Widgets are listed and described in B “List of Widgets” on page 321.

Most widgets are single items that can be added from the tool bar onto a **Tile** or **Schematic**. Once added, they can be resized and repositioned. These ‘non-shape’ widgets include:

- Tally Lamp.
- Clock.
- Audio Bars.

There are some ‘shape’ widgets that require ‘drawing’ to form their shape when they are added to a **Tile** or **Schematic**. These widgets include:

- Line.
- Rectangle.
- Ellipse.
- Triangle.
- Polygon.

The ‘drawing’ of each these shape widgets when being added to a **Tile** or **Schematic** is described in B “List of Widgets” on page 321.

3.5.2 Add a Widget to a Tile or Schematic

To add a widget to a **Tile** or **Schematic**:

Adding a Single Widget:

1. Click on the corresponding widget icon in the **Widgets Tool Bar**. This selects the widget item to add and enters ‘single-widget’ mode.

2. Move your cursor onto the **Tile** or **Schematic**.
A widget is attached to your cursor.
3. Position the widget on the **Tile/Schematic** and click to release the widget.

The widget has been added to the **Tile** or **Schematic**. Editing of widgets can now be done. For example, widgets can be repositioned or resized.

Press the '**Esc**' key on the keyboard to exit 'single-widget' mode at any time to abort adding a widget.

Adding Multiple Widgets:

1. Hold down the 'Ctrl' key on the keyboard.
2. Click on the corresponding widget icon in the **Widgets Tool Bar**.
This selects the widget item to add and enters 'multiple-widget' mode.
3. Move your cursor onto the **Tile** or **Schematic**.
A widget is attached to your cursor.
4. Position the widget on the **Tile/Schematic** and click to release a widget.
Another widget is attached to the cursor.
5. Position as many widgets as required.
6. For the last widget, release the '**Ctrl**' key before clicking to place the widget.

Multiple widgets have been added to the **Tile** or **Schematic**. Editing can now be done.

Right-click to deselect the widget and exit 'multiple-widget' mode at any time, to stop adding widgets.

Shape widget items:

For adding 'shape' widget items,
see the corresponding shape widget item in B "List of Widgets" on page 321.

3.5.3 Reposition/Resize Widgets

After a widget is added to the **Tile** or **Schematic** it may be repositioned, resized or have its aspect ratio set:

1. Click on the widget to select it.
Properties of the selected widget are shown in the **Properties Box**.

Repositioning:

2. Reposition the widget (see Figure 86) by:
 - Holding down the left-mouse button and dragging the widget to reposition it.Or
 - Changing the widget position properties (**Position > X** and **Position > Y**) in the **Properties Box**.

Click and Drag widget to reposition

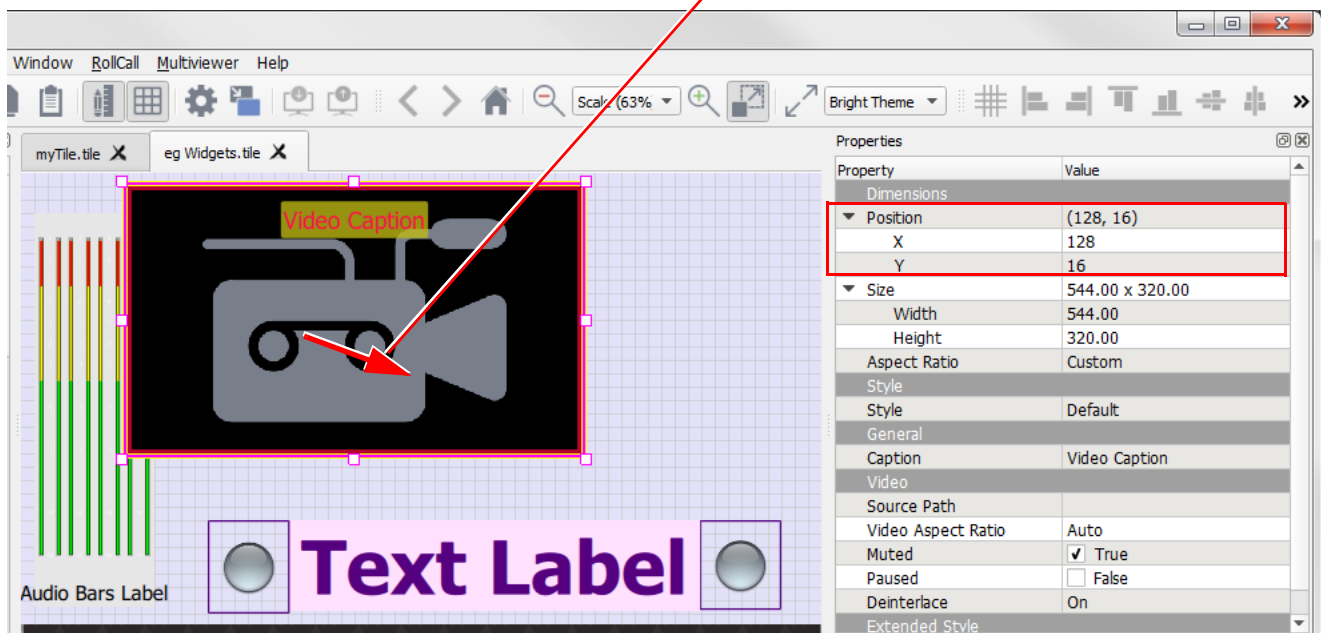
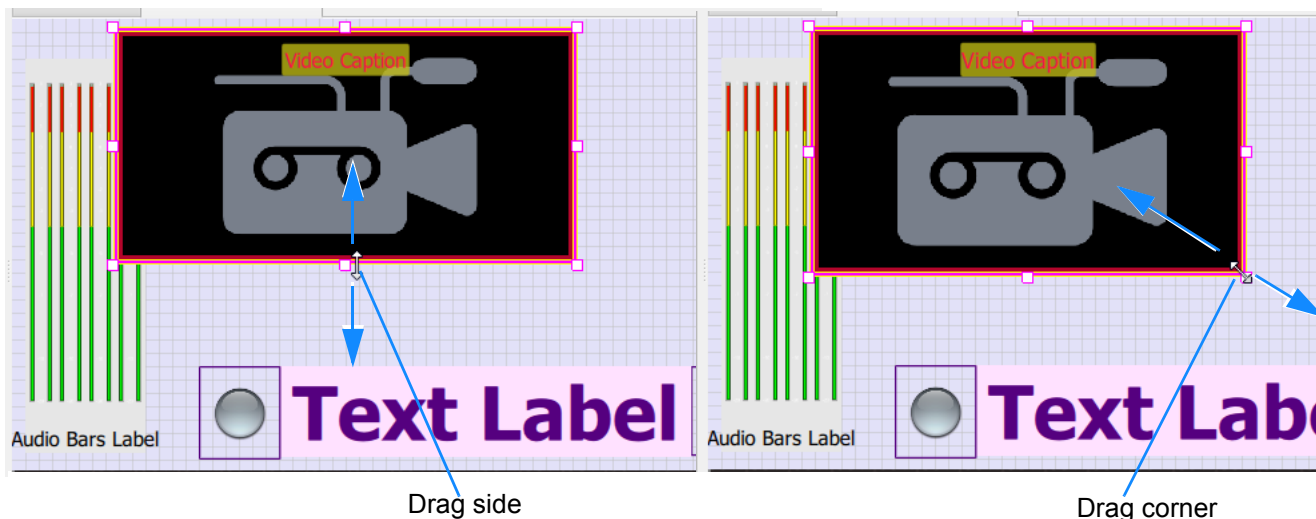


Figure 86 Repositioned Widget

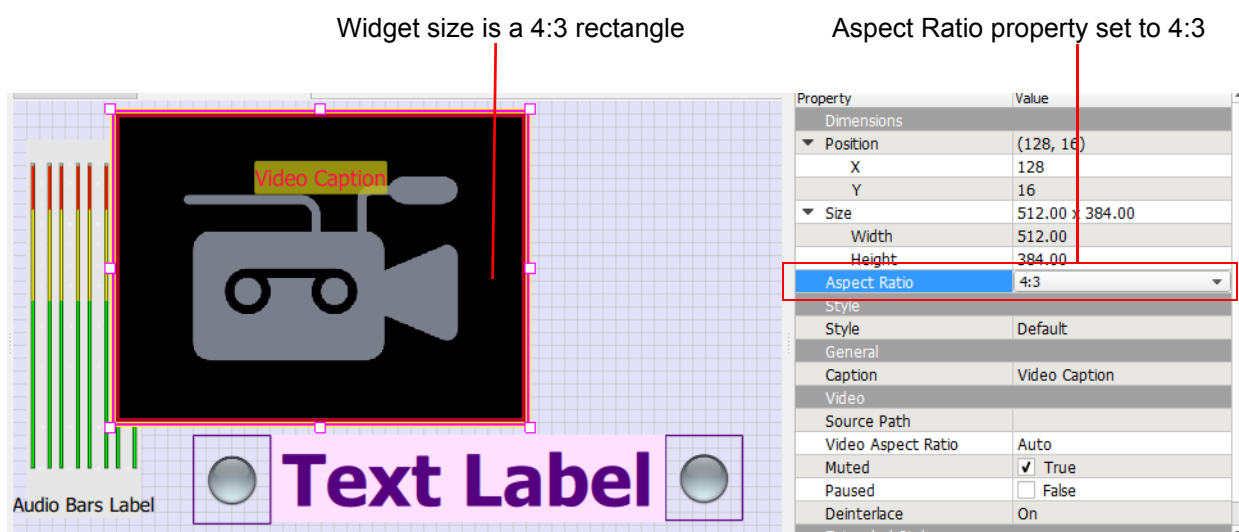
Resizing:

3. Change the size of a selected widget (see Figure 86) by:
 - Dragging the widget edges and corners to resize it.
 Or
 - Changing the widget size properties (**Size > Width** and **Size > Height**) in the **Properties Box**.

**Figure 87 Resize a Widget****Aspect Ratio:**

For a Video widget, the aspect ratio can be set with the **Dimensions > Aspect Ratio** property:

4. Select the widget **Aspect Ratio** property and set it via the Aspect Ratio drop-down box.

**Figure 88 Widget Aspect Ratio**

3.6 Schematic 'Right-Click' Context Menus

Various operations may be carried out on the graphical widget object(s) in the **Tile Editor** schematic area via a 'right-click'. A right-click will display a pop up menu.

Operations that may be carried out from these right-click menus include:

- Go back / forward between tabs.
- Selection.
- Cut/ Copy/ Paste.
- Deleting.
- Grouping.
- Aligning.
- Layer Ordering.
- Editing Behaviours.

Right-clicking in the **Tile Editor** can be done:

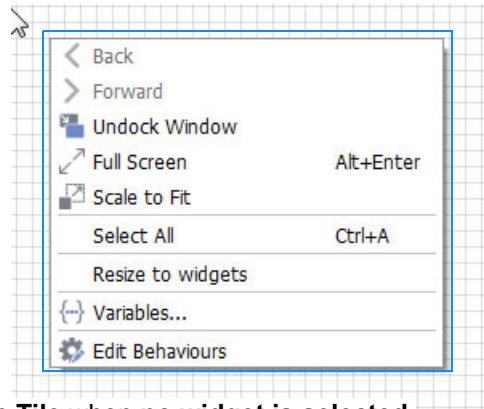
- a On the tile, with no widgets selected.
- b With one widget selected.
- a With more than one widget selected.

These right-click operations bring up similar pop up menus, see Figure 89a, b and c.

Table 13, Table 14 and Table 15 briefly describe the right-click menu items for each of the cases on.

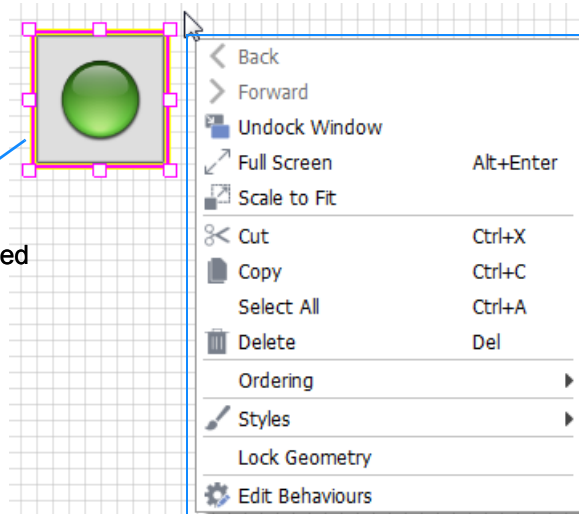
Figure 89.

No widget selected



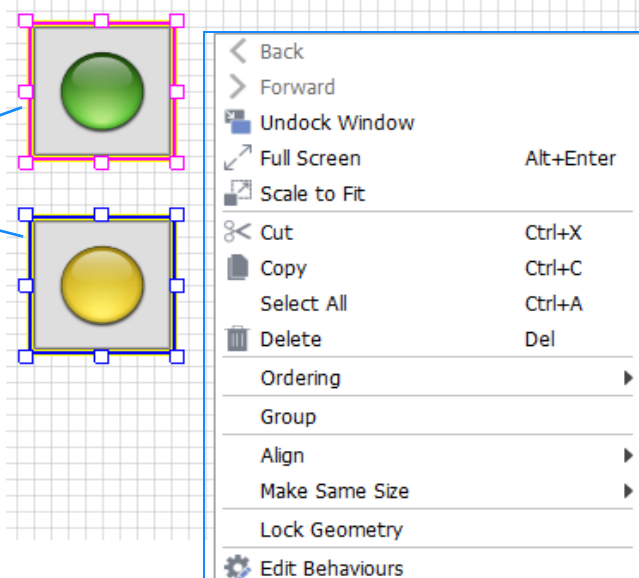
a) 0, Right-Click on Tile when no widget is selected

One widget selected



b) 1, Right-Click pop-up menu when one widget is selected.

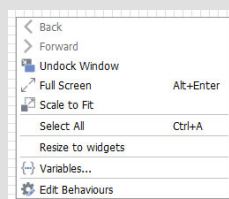
More than one widget selected



c) >1, Right-Click pop-up menu when more than one widget is selected

Figure 89 Pop up Menu. Right-Click on Tile Editor Schematic Area:

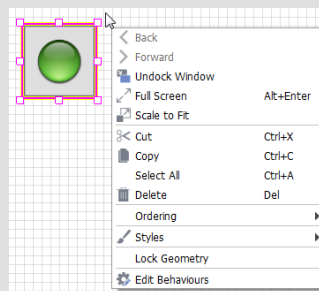
- a) 0, when no widget selected.
- b) 1, when one widget is selected.
- c) >1, when more than one widget is selected.

Right-Click on **Tile** when **no widget** is selected

| Menu Item | Description |
|--|--|
| Back | Go back to previous tab. |
| Forward | Go forward to next tab. |
| Undock Window / Dock Window | Click to place tab into a floating window. / Click to re-dock a floating window to tab. |
| Full Screen | Click to toggle between full-screen mode and normal viewing in a tab window. Also Alt-Enter . |
| Scale to Fit | Scale schematic to fit schematic area. Click Scale to Fit menu item to toggle the 'scale to fit' feature on/off. |
| Select All | Select all widgets on the tile. Also Ctrl-A . |
| Resize to Widgets | Resize the tile or schematic to the minimum size necessary for all the widgets. |
| Variables... | Click to define user Tile Variables . Tile Variables are similar to user project variables but they only apply to a tile. |
| Edit Behaviours | Click to define a certain behavior patterns for use by widgets, for example, to enable on-screen notifications via some binding. For example, a behavior may be defined to monitor certain log messages on a log server. See Section 3.8 "Auto Fill Facility" on page 101. |

Table 13 Tile Editor Right-Click Menu Items - No widgets selected

Right-Click pop-up menu when **one widget** is selected.



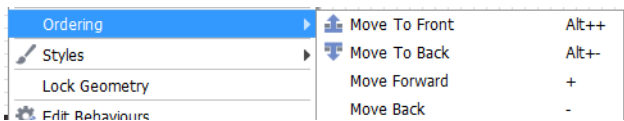
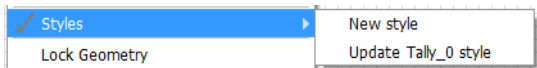
| Menu Item | Description |
|------------------------------------|--|
| Back | Go back to previous tab. |
| Forward | Go forward to next tab. |
| Undock Window / Dock Window | Click to place tab into a floating window. / Click to re-dock a floating window to tab. |
| Full Screen | Click to toggle between full-screen mode and normal viewing in a tab window. Also Alt-Enter . |
| Scale to Fit | Scale schematic to fit schematic area. |
| Cut | Cut selected graphical object from tile or schematic. |
| Copy | Copy selected graphical object from tile or schematic. |
| Paste | Paste copied a graphical object(s) onto a tile or schematic. |
| Select All | Select all widgets on the tile. Also Ctrl-A . |
| Delete | Delete selected graphical object(s) from a tile or schematic. |
| Ordering | Move selected object(s) layering [Short cut keystroke is in brackets]:  <ul style="list-style-type: none"> • To Front. [Alt +]. • To Back. [Alt -]. • Forward. [+]. • Back Also [-]. |
| Styles | Click to use the selected widget in a widget style:  <ul style="list-style-type: none"> • New Style - Create a new widget style from the widget on the tile or schematic. • Update "Style-Name" style - Update an existing widget style from the widget on the tile or schematic. |

Table 14 Tile Editor Right-Click Menu Items - One Widget selected

Lock Geometry
Unlock Geometry

Click to lock / unlock a widget on a **Tile** or a **Schematic**.

When a widget is locked in this way, its position, size and aspect ratio cannot be changed by dragging or by changing properties.

A padlock symbol appears on a selected widget if it is locked and its corresponding properties are grayed out in the **Properties Box**.



| Properties | |
|--------------|---------------|
| Property | Value |
| Dimensions | |
| Position | (190, 0) |
| X | 190 |
| Y | 0 |
| Size | 60.00 x 60.00 |
| Width | 60.00 |
| Height | 60.00 |
| Aspect Ratio | Custom |

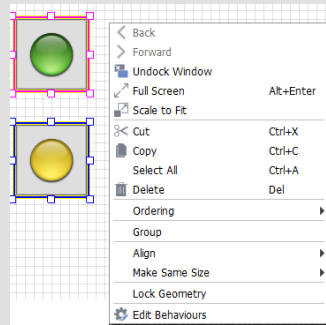
Edit Behaviours

Click to open the **Behaviours and Bindings Editor**.

Do this to define a certain behavior pattern for use by widgets, for example, to enable on-screen notifications via some binding.

See Section 5 “Widget Behaviours and Bindings” on page 130.

Table 14 Tile Editor Right-Click Menu Items - One Widget selected (Continued)

Right-Click pop-up menu when **more than one widget** is selected

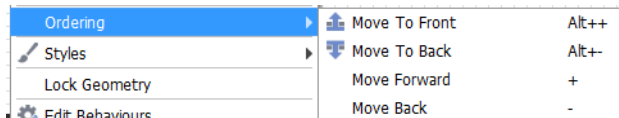
| Menu Item | Description |
|------------------------------------|--|
| Back | Go back to previous tab. |
| Forward | Go forward to next tab. |
| Undock Window / Dock Window | Click to place tab into a floating window. / Click to re-dock a floating window to tab. |
| Full Screen | Click to toggle between full-screen mode and normal viewing in a tab window. Also Alt-Enter . |
| Scale to Fit | Scale schematic to fit schematic area. |
| Cut | Cut selected graphical object from tile or schematic. |
| Copy | Copy selected graphical object from tile or schematic. |
| Paste | Paste copied a graphical object(s) onto a tile or schematic. |
| Select All | Select all widgets on the tile. Also Ctrl-A . |
| Delete | Delete selected graphical object(s) from a tile or schematic. |
| Ordering | Move selected object(s) layering [Short cut keystroke is in brackets]:  <ul style="list-style-type: none"> • To Front. [Alt +]. • To Back. [Alt -]. • Forward. [+]. • Back Also [-]. |
| Group / Ungroup | Click to: <ul style="list-style-type: none"> • Group together selected widgets. • Ungroup grouped widgets. <p>Using Groups, the on-screen layout of widgets can be controlled with Group Layouts.</p> <p>See Section 6.1 “Widget Groups” on page 147 for more information on Groups. See Section 6.3 “Group Layout” on page 160 for more information on Group Layouts.</p> |

Table 15 Tile Editor Right-Click Menu Items - More Than One Widget selected

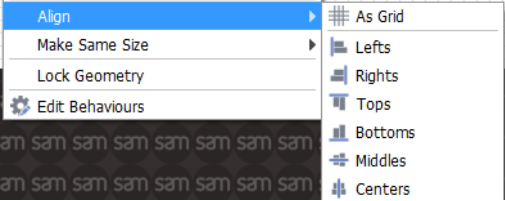
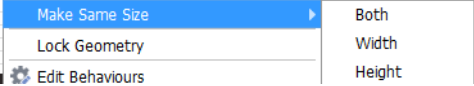

| | |
|----------------------------------|---|
| Align | Align selected widgets: <div></div> <ul style="list-style-type: none">Align as Grid.Left sides.Rights sides.Top sedges.Bottom edges. |
| Make Same Size | Make selected widgets the same size: <div></div> <ul style="list-style-type: none">Width and Height.Width only.Height only. |
| Lock Geometry Unlock Geometry | <p>Click to lock /unlock a widget's geometry when resizing.</p> <p>A padlock symbol appears on a selected widget if it is locked.</p> <div></div> |
| Edit Behaviours | <p>Click to define a certain behavior patterns for use by widgets, for example, to enable on-screen notifications via some binding.</p> <p>For example, a behavior may be defined to monitor certain log messages on a log server.</p> <p>See Section 5 “Widget Behaviours and Bindings” on page 130.</p> |

Table 15 Tile Editor Right-Click Menu Items - More Than One Widget selected (Continued)

3.7 Tile Editor - Advanced Tool Bar



The **Tile Editor** screen contains an **Advanced Tool Bar** to the lower right of the schematic area, see Figure 90.

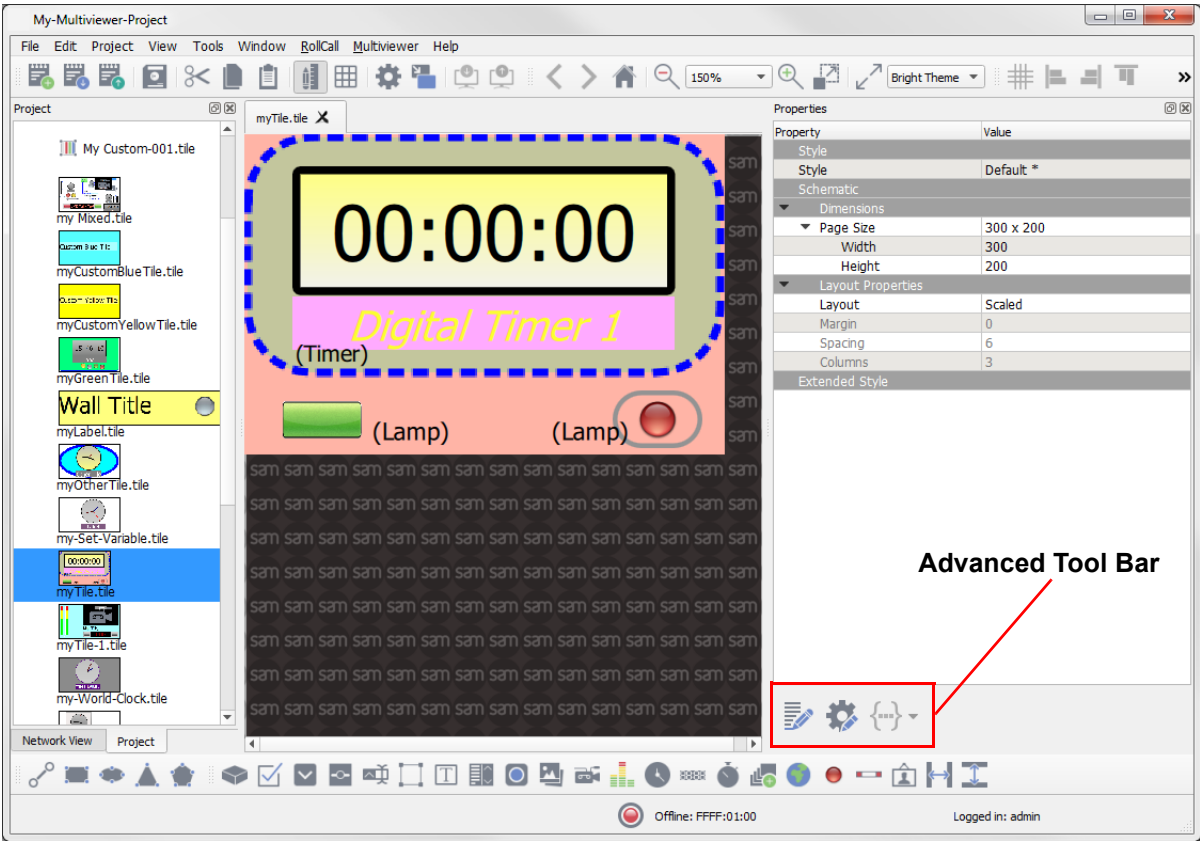


Figure 90 Tile Editor Advanced Tool Bar

Figure 91 shows the **Advanced Tool Bar**. The icons are briefly described in Table 16.



Figure 91 Advanced Tool Bar




| Item | Description |
|---|--|
| Auto Fill  | <p>Click icon to fill out the text properties of a set of widgets.</p> <p>This is aimed at speeding up the configuration of multiple widgets on a Tile.</p> <p>For example, if a group of buttons are required to be named in sequence.</p> <p>See Section 3.8 “Auto Fill Facility” on page 101.</p> |
| Edit Behaviours  | <p>Click icon to define and edit behavior patterns used by widgets. The ‘Behaviour and Bindings’ graphical editor is opened.</p> <p>A Behaviour gets information from system status properties.</p> <p>A Orbit Behaviour may be defined to get or to set system status information For example:</p> <ul style="list-style-type: none"> • Extract certain system information from log messages from a log server. (For example, presence of a video input signal.) • Get audio levels for an input. <p>The system information carried by a Behaviour may then be used in a Orbit Binding to apply values to a widget property in order to carry out some action. (For example, to provide on-screen notifications, display an audio level, assert a GPI output.)</p> <p>See Section 5 “Widget Behaviours and Bindings” on page 130.</p> |
| Set Variables  | <p>Icon is active when a widget property is selected.</p> <p>Click icon to assign a Orbit local variable to a property value. A Local variable applies to the tile (i.e. it has tile scope.). The local variables can then be changed at wall level when the tile is used on a wall.</p> <p>For example, a UMD widget: a UMD tile may be defined once and its properties can be changed wherever it is used at the wall level (for example, TSL address).</p> <p>See Section 7.2.2 “Set Variables Icon and User Tile Variables” on page 178.</p> |

Table 16 Tile Editor - Advanced Tool Bar Items

3.8 Auto Fill Facility

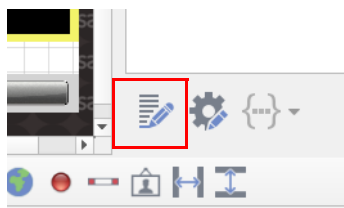


Figure 92 Advanced Tool Bar - Auto Fill Icon

The **Auto Fill** icon in the **Tile Editor's Advanced Tool Bar** allows properties of a selection of widgets to be filled out quickly. The **Auto Fill** dialog is shown when a group of widgets are selected and the **Auto Fill** icon is clicked. See Figure 93.

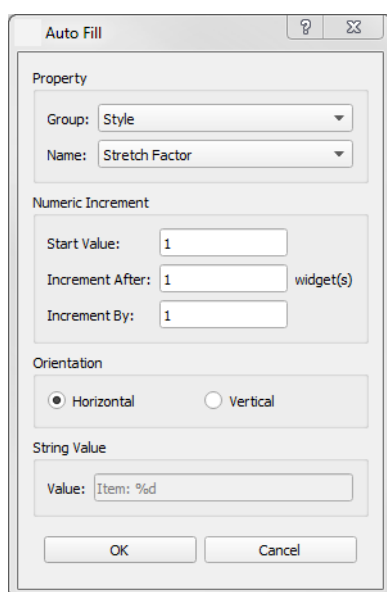


Figure 93 Auto Fill Dialog

| Item | Description |
|---------------------------|--|
| Property: | |
| Group | Drop-down box. Select widget property group. Selects the set of properties that may be auto-filled, which are then selectable with the Name drop-down box. |
| Name | Drop-down box. Select the widget property to have its value filled out automatically. |
| Numeric Increment: | |
| Start Value | Select starting value for widget numbering. |
| Increment After | Select when to increment, i.e. numbering does not increment for this many widgets. |
| Increment By | Select increment value. |
| Orientation: | |
| Horizontal | Select renumbering horizontally, then vertically. |
| Vertical | Select renumbering vertically, then horizontally. |
| String Value: | |
| Value | Property text value to auto fill with. Typically set to be text with a "%d". E.g. "Channel %d" or "INPUT %d" Where "%d" is a placeholder representing the auto fill renumbering value. |

Table 17 Auto Fill Dialog items

Note:**Selection/ Deselection of graphical items when editing items in the Tile Editor:**

| Select / Deselect | Action |
|------------------------------|---|
| Deselect All | Click outside the schematic area. Or press the ' Esc ' keyboard key. |
| Select | Click on an item to select it. Alt-Click selects another item: <ul style="list-style-type: none"> • In a stack of items. • Or selects the Group Layout that an item belongs to. |
| Add Select | Shift-Click on an item to add it to current selection. Shift-Alt-Click to add another item from a stack of items to the current selection. |
| Remove from Selection | Shift-Click on an <i>already-selected</i> item to remove it from the current selection. |
| Select All | 'Ctrl-A' selects all items. |
| Select an Area | Shift drag cursor on screen to describe a rectangular area. All objects wholly or partly in the area will be selected. |

3.8.1 Example - Labeling Widgets in a Numbered Sequence.

A group of labels are required to be named in a numbered sequence:

1. Select a group of widgets in the **Tile Editor**. See Figure 94.



Figure 94 Advanced Tool Bar - Auto Fill Icon

2. Click **Auto Fill** in the **Advanced Tool Bar**.
The **Auto Fill** dialog is shown. See Figure 95.

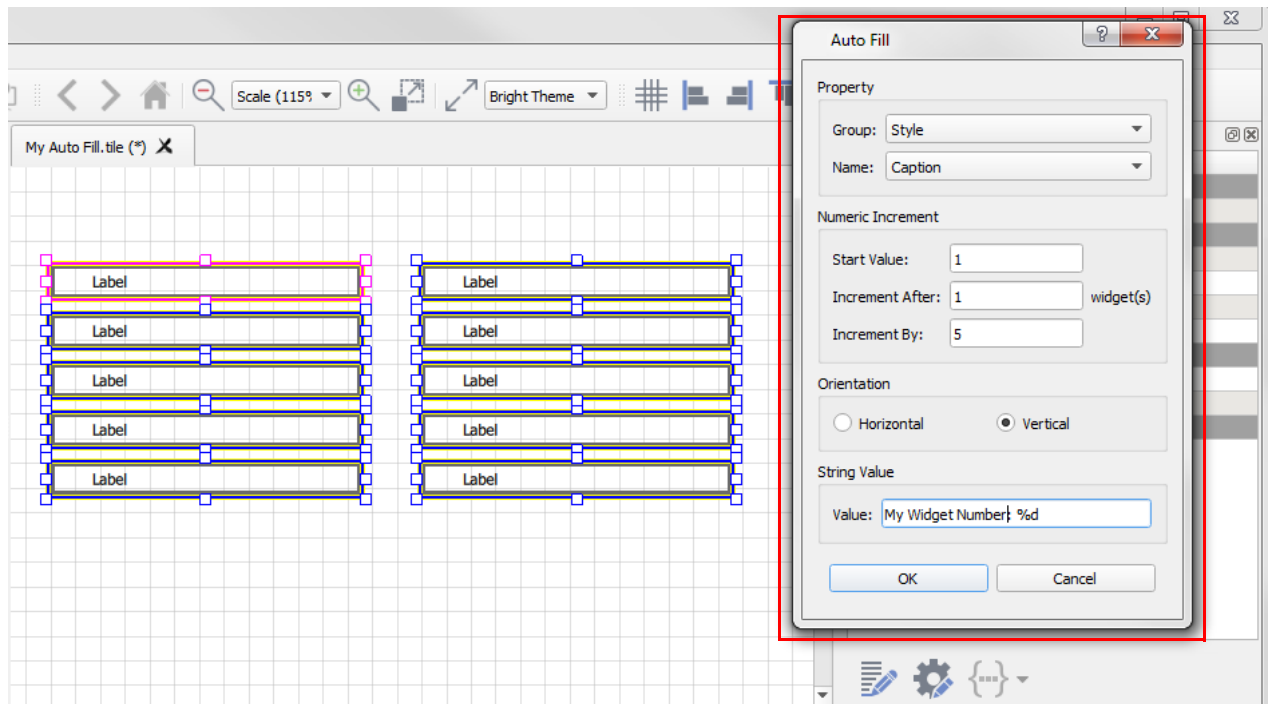


Figure 95 Auto Fill Dialog

3. Select 'Caption' in the **Name** item,
4. Set **Start Value** to 1.
Set **Increment After** to 1.
Set **increment By** to 5.
5. Select **Orientation** Vertical.
6. Enter "My Widget Number: %d" in the **Value** item.

Note:

Value: item "%d":

"%d" in the **Value:** item is a placeholder for the incrementing number.

7. Click **OK**.
The label widget caption text is then filled out to be:
"My Widget Number: 1", "My Widget Number: 6", "My Widget Number: 11", etc.
See Figure 96.

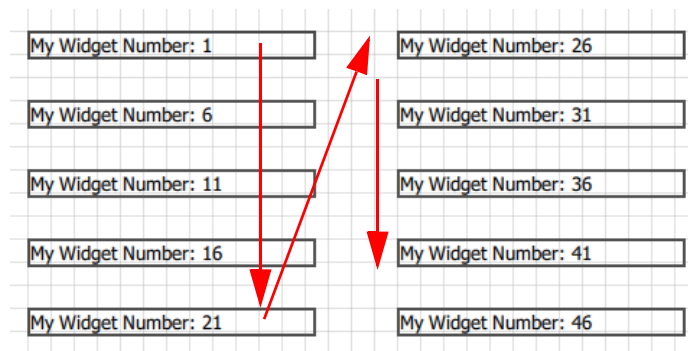


Figure 96 Auto Fill Example

The labels' captions have been filled out with text that contains an incrementing numeric part. Primarily, numbering increments in a vertical direction and then, secondarily, horizontally.

Increment After:

Figure 97 shows the effect of using the 'Increment After' item in the **Auto Fill** dialog.

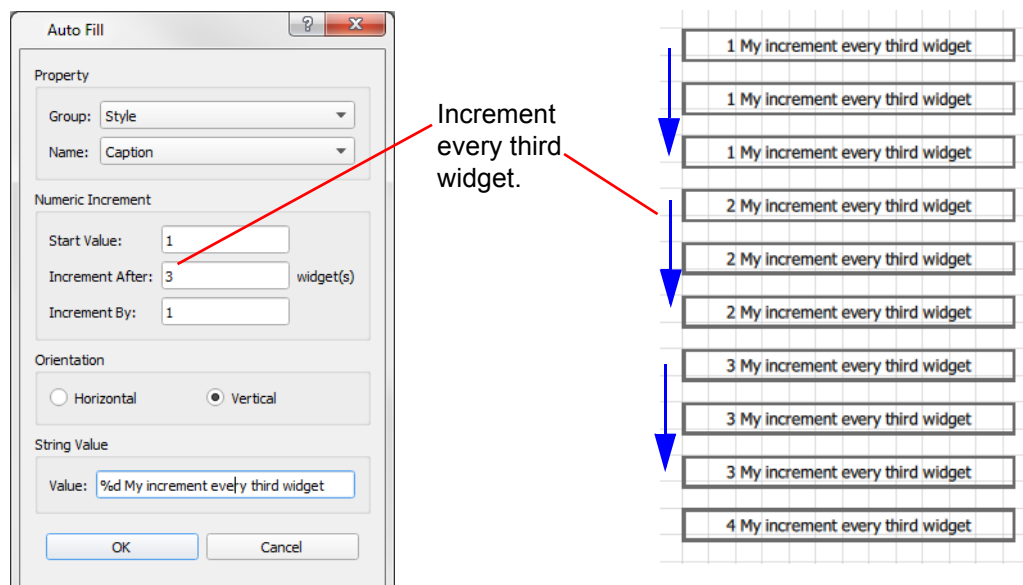


Figure 97 Auto Fill Example, showing effect of “Increment After” setting

Increment By:

Figure 98 shows the effect of using the 'Increment By' item in the **Auto Fill** dialog.

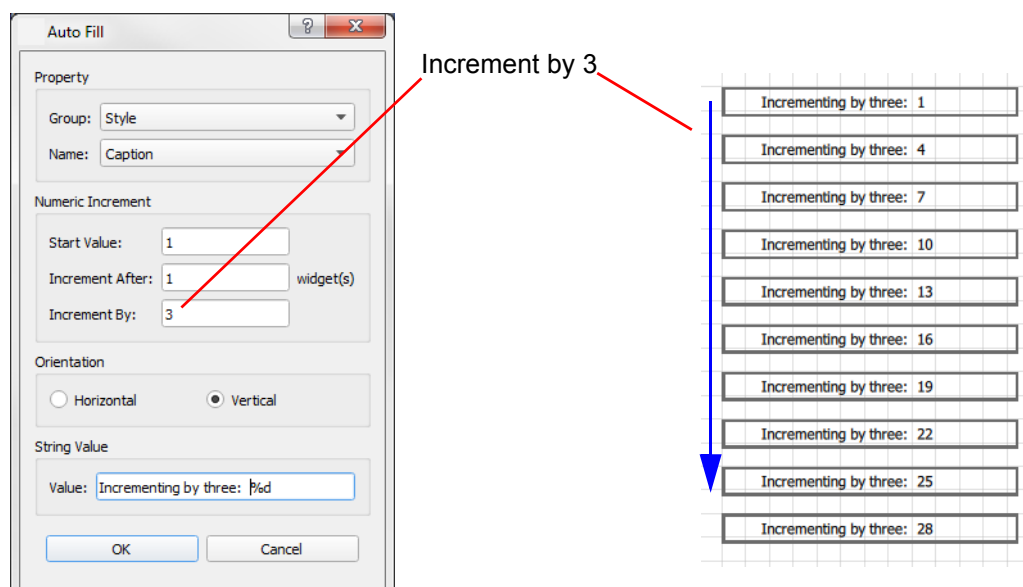


Figure 98 Auto Fill Example, showing effect of “Increment By” setting

3.9 Auto-Tiles for Multiviewers

An **Auto-Tile** is a multiviewer video wall tile which is generated automatically by the Orbit software. Its properties and functionality are generated and then maintained by the Orbit software.

An **Auto-Tile** can be added to a wall in the **Wall Editor** in its **Tile Grid** mode (see Section 2.6 “Wall Editor - Tile Grid Mode” on page 31).

An **Auto-Tile** contains a fixed set of graphical widgets and is designed to address common use-cases. For example, a video **Auto-Tile** comprises a video source, a UMD and an audio bars widget. Various widget properties may be edited on the **Auto-Tile**.

In an **Auto-Tile**, the functionality of the widgets is defined “under the hood” with Orbit **Behaviours** and **Bindings**. Widget properties on the **Auto-Tile** can be modified by the user to modify the widget behavior without resorting to editing **Behaviours** or **Bindings**. For example, the number of audio bars displayed can be set by editing properties.

Note:

Auto-Tile and Custom Tiles:

An **Auto-Tile** is different from a **Custom Tile**.

An **Auto-Tile** is a pre-built widget. It offer some properties for a user to be able to modify some functionality.

A **Custom Tile** is created by a user; it can be created from an empty tile or created from an **Auto-Tile** (recommended). A **Custom Tile** can hold any combination of widgets and its functionality can be fully tailored and defined.

There are different types of **Auto-Tile** for common multiviewer tile requirements:

- Video tile for 3G and HD video.
- Video tile for 4K video.
- Analogue clock tile.
- Digital clock tile.

The type of **Auto-Tile** is set with a tile property.

To select an **Auto-Tile** type:

1.
- In the **Wall Editor**, select an **Auto-Tile** in a wall with just a with a **single-click**.
The widget's properties are shown in the **Properties Box**. See Figure 99.

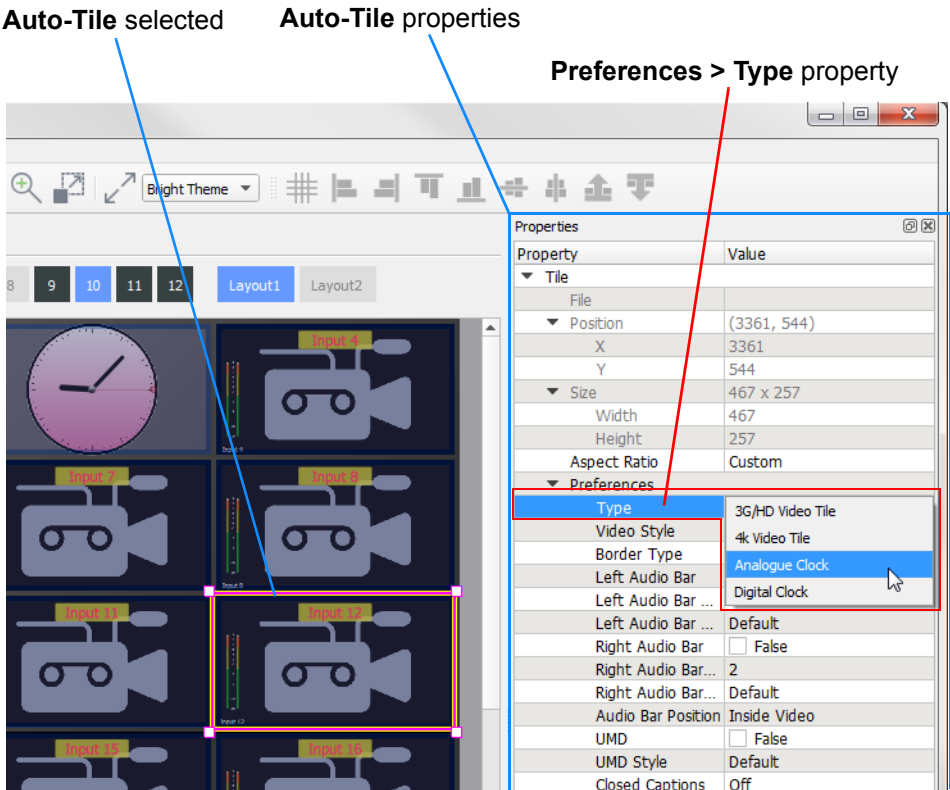


Figure 99 Auto-Tile Type Property

2.
- The **Preferences > Type** property may be changed in the **Properties Box**.
Table 18 lists **Auto-Tile** types.

| Auto-Tile Type | Description |
|------------------|---|
| 3G/HD Video Tile | Video tile using one multiviewer input. See “Video Auto-Tiles - 3G/HD and 4K” on page 107. |
| 4k Video Tile | Video tile using four multiviewer inputs. These four inputs do not have to be adjacent inputs. For example, inputs 6, 7, 12 and 13 may be used. See “Video Auto-Tiles - 3G/HD and 4K” on page 107. |
| Analogue Clock | Tile showing an analogue clock face. See “Clock Auto-Tiles” on page 111. |
| Digital Clock | Tile showing a digital clock face. See “Clock Auto-Tiles” on page 111. |

Table 18 Auto-Tile Types

Auto-Tiles have other properties that affect the widgets in them. These properties are shown in the **Properties Box** when the **Auto-Tile** is selected in the wall. These properties can be edited in the **Properties Box**.

Note: *Single-clicking* on an **Auto-Tile** in a wall:

This selects the **Auto-Tile** in the **Wall Editor**.
This shows the tile's properties in the properties box.

Changes can be made to the properties of the **Auto-Tile** in the wall.
The resulting tile is still an **Auto-Tile**.

Importantly, however, changes made in the **Tile Editor** result in a **Custom Tile**.

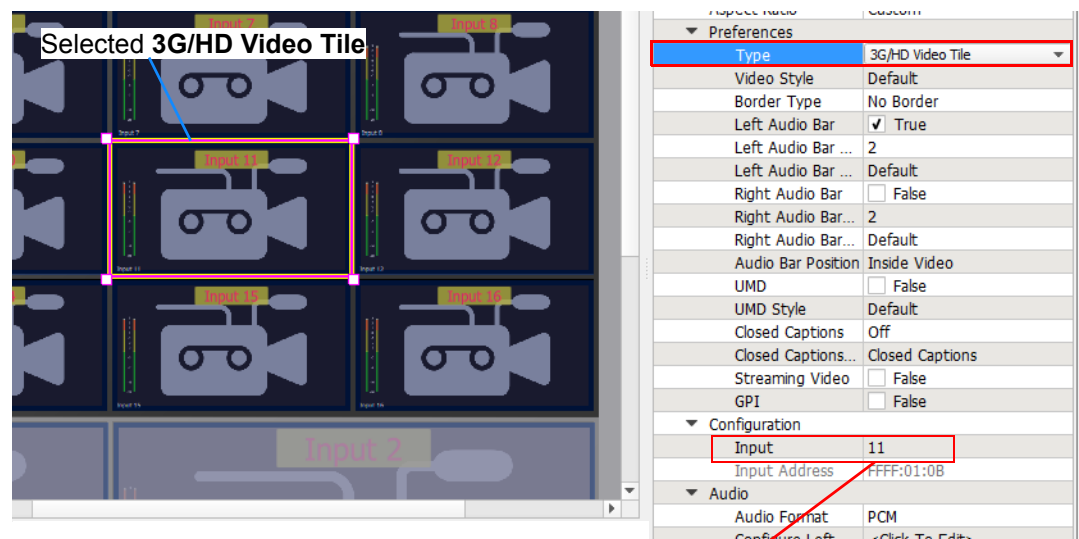
Note: *Double-clicking* on an **Auto-Tile** in a wall:

This prompts to create a new **Custom Tile** from the **Auto-Tile** and opens the new tile in the **Tile Editor**. A **Custom Tile** is the result.

3.9.1 Video Auto-Tiles - 3G/HD and 4K

The 3G/HD and 4K video **Auto Tiles** are similar. The 4K version accepts four multiviewer (3G) inputs to make a 4K input and these are specified in the 4K tile properties. See Figure 100 and Figure 101.

In particular, video **Auto-Tiles** have widget styles and features that can be turned on/off and positioned on-screen by simply changing some of the **Auto-Tile** properties: The properties are changed at wall-level.



Multiviewer input 11 shown assigned to the tile.

Figure 100 3G/HD Video Auto-Tile

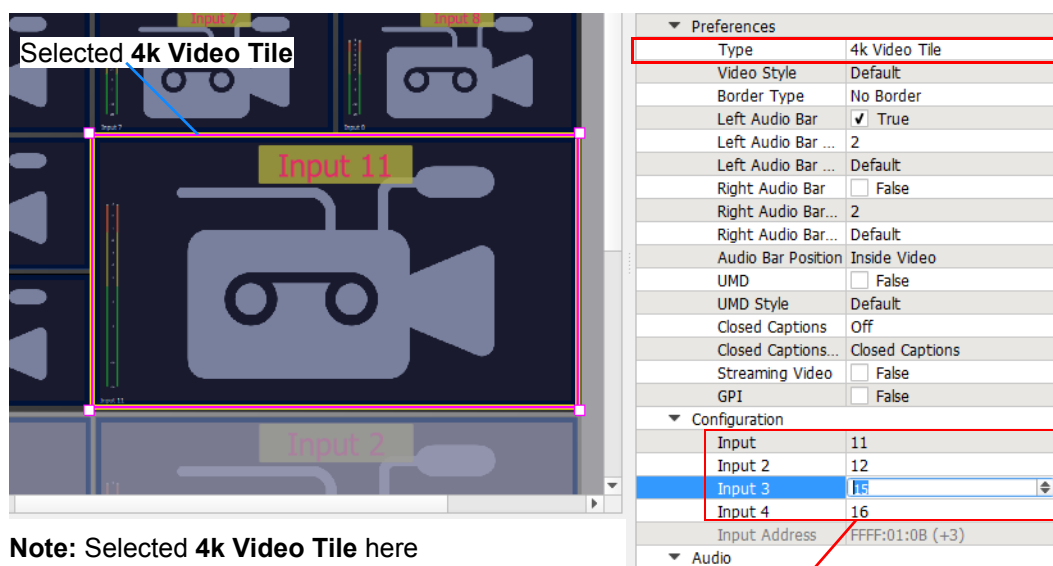


Figure 101 4k Video Auto-Tile

3.9.2 Video Auto-Tile - Style Properties

The on-screen appearance (style) each of the video **Auto-Tile** can be changed by selecting the style of its constituent widgets via the **Auto-Tile**'s style properties. See Figure 102.

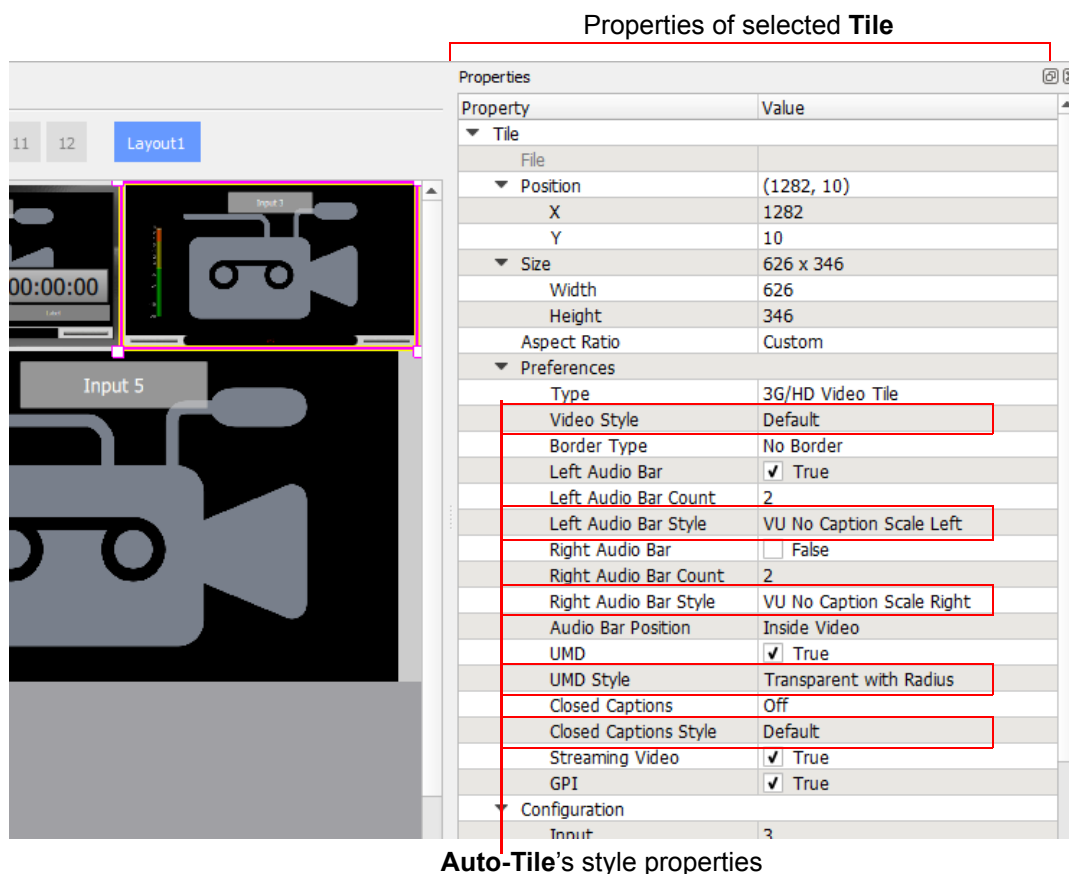
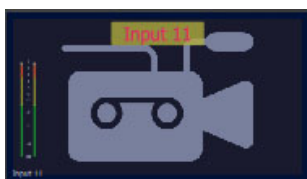


Figure 102 Tile Style Properties

3.9.3 Video Auto-Tile - Properties

Figure 103 shows the video **Auto-Tile** properties when the tile is selected on a wall.

| Property | Value |
|--------------------|--|
| ▼ Tile | |
| File | |
| ▼ Position | (2884, 544) |
| X | 2884 |
| Y | 544 |
| ▼ Size | 467 x 257 |
| Width | 467 |
| Height | 257 |
| Aspect Ratio | Custom |
| ▼ Preferences | |
| Type | 3G/HD Video Tile |
| Video Style | Default |
| Border Type | No Border |
| Left Audio Bar | <input checked="" type="checkbox"/> True |
| Left Audio Bar ... | 2 |
| Left Audio Bar ... | Default |
| Right Audio Bar | <input type="checkbox"/> False |
| Right Audio Bar... | 2 |
| Right Audio Bar... | Default |
| Audio Bar Position | Inside Video |
| UMD | <input type="checkbox"/> False |
| UMD Style | Default |
| Closed Captions | Off |
| Closed Captions... | Closed Captions |
| Streaming Video | <input type="checkbox"/> False |
| GPI | <input type="checkbox"/> False |
| ▼ Configuration | |
| Input | 11 |
| Input Address | FFFF:01:0B |
| ▼ Audio | |
| Audio Format | PCM |
| Configure Left ... | <Click To Edit> |
| Configure Right... | <Click To Edit> |



a) Video **Auto-Tile** properties

| Property | Value |
|--------------------|--|
| ▼ Tile | |
| File | |
| ▼ Position | (2884, 544) |
| X | 2884 |
| Y | 544 |
| ▼ Size | 467 x 257 |
| Width | 467 |
| Height | 257 |
| Aspect Ratio | Custom |
| ▼ Preferences | |
| Type | 3G/HD Video Tile |
| Video Style | Default |
| Border Type | No Border |
| Left Audio Bar | <input checked="" type="checkbox"/> True |
| Left Audio Bar ... | 2 |
| Left Audio Bar ... | Default |
| Right Audio Bar | <input checked="" type="checkbox"/> True |
| Right Audio Bar... | 2 |
| Right Audio Bar... | Default |
| Audio Bar Position | Inside Video |
| UMD | <input checked="" type="checkbox"/> True |
| UMD Style | Default |
| Closed Captions | Bottom |
| Closed Captions... | Closed Captions |
| Streaming Video | <input checked="" type="checkbox"/> True |
| GPI | <input checked="" type="checkbox"/> True |
| ▼ Configuration | |
| Input | 11 |
| Input Address | FFFF:01:0B |
| ▼ Audio | |
| Audio Format | PCM |
| Configure Left ... | <Click To Edit> |
| Configure Right... | <Click To Edit> |
| ▼ UMD | |
| UMD Protocol | Version 5.0 |
| 5.0 Screen Index | 0 |
| 5.0 Display Index | 0 |
| ▼ Streaming | |
| Source Path | rtsp://10.162.51.31:5986... |
| ▼ GPI | |
| GPI Mode | Input |
| GPI Port Index | 1 |
| ▼ Closed Captions | |
| Page | 100 |



b) Video

Auto-Tile properties when the following **Auto-Tile** features are enabled:

- Right Audio Bars
- UMD
- Closed Caption
- Streaming Video
- and GPI.

Figure 103 Video Auto-Tile Properties (Auto-Tile selected on a Wall):

a) Video Auto-Tile properties.

b) Properties when extra Auto-Tile features are enabled.

3.9.4 Video Auto-Tile - Appearances

Figure 104 shows some different appearances of the video **Auto-Tile** when configured with various properties set.

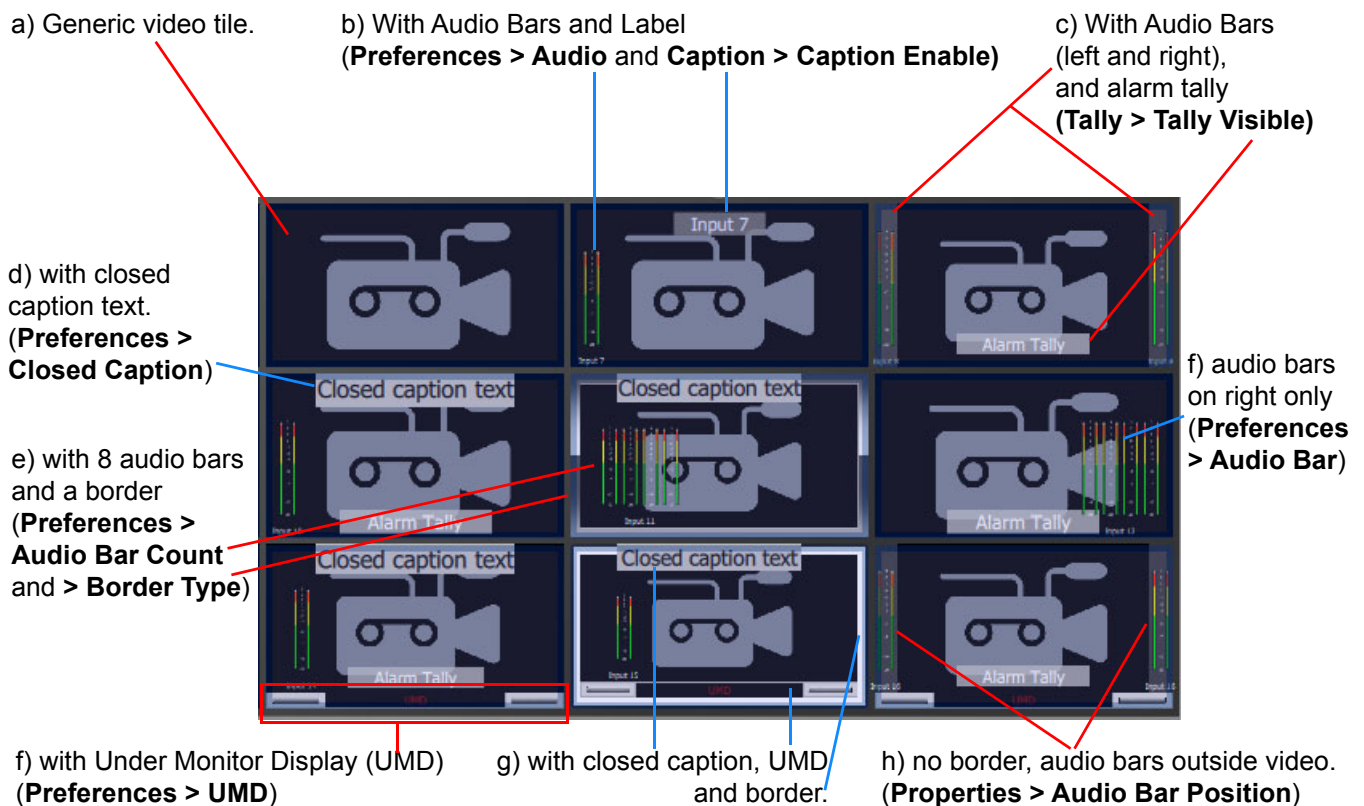


Figure 104 Video Auto-Tiles

The video widget, Audio widget, UMD widget and GPI can also be configured to look at various data sources for displaying data: Table 19 lists some of the properties involved.

| Feature | Property |
|--|---|
| Video Input source for the Video widget. | Configuration > Input. |
| Video Streaming (H.264) source for Video widget. | Preferences > Streaming Video, Streaming > Source Path. |
| Audio source the Audio widget audio bars. | Audio > Configure Left Bars, Audio > Configure Right Bars. |
| Audio format for the Audio widget audio bars. | Audio > Audio Format, |
| UMD protocol for UMD widget. | Preferences > UMD, UMD > UMD Protocol. |
| Which Multiviewer GPI to use in a tile. | Preferences > GPI, GPI > GPI Port Index. |
| GPI In or GPI Out function for the tile. | Preferences > GPI, GPI > GPI Mode. |
| Closed Captions | Preferences > Closed Captions, Preferences > Closed Captions Style. |
| Tally Text | Theme Editor (Video widget style): Tally > Text, Tally > Tally Visible. |
| Video Caption | Theme Editor (Video widget style): Caption > Caption Enable |
| Style of widget | Preferences > Video Style |

Table 19 Some Properties that configure Video Auto-Tiles

3.9.5 Clock Auto-Tiles

Analog Clock:

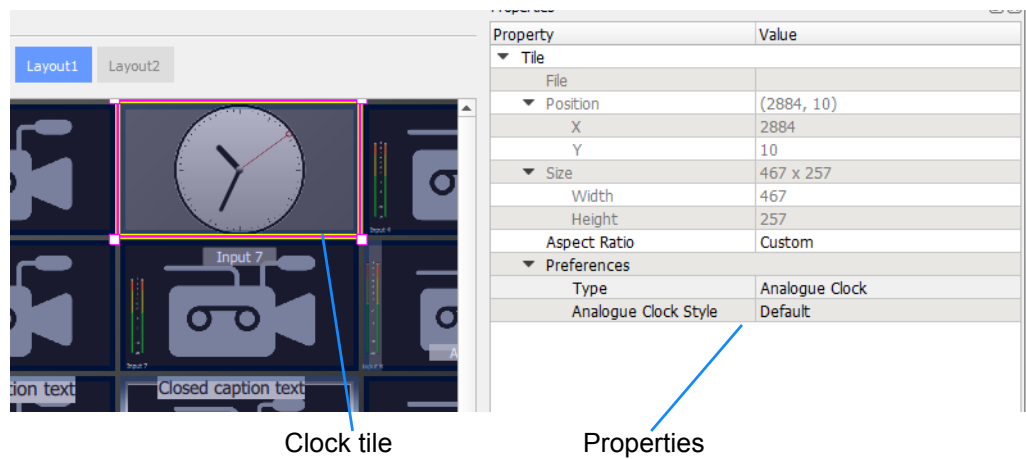


Figure 105 Analogue Clock Auto-Tile

Digital Clock:

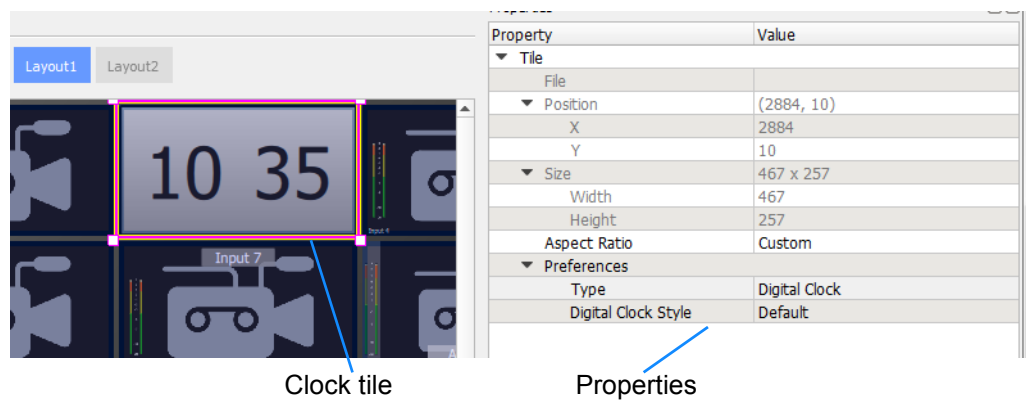


Figure 106 Digital Clock Auto-Tile

The Analogue and Digital Clock Auto-Tiles have similar sets of properties. The main clock properties are in the **Themes Editor**, where clock styles are defined. (See Section 4 “Theme Editor - Widget Styles” on page 112.)

Table 19 lists some of the properties involved in configuring the clocks.

| Feature | Property |
|------------|--|
| Clock face | Theme Editor: Clock > Clock Face Color, Font; |
| Clock time | Theme Editor: Clock > Display Seconds and Display 24 Hour Clock. Clock > Clock Source Type and Offset Hours |

Table 20 Some Properties that configure Video Auto-Tiles

4

4 Theme Editor - Widget Styles

Theme Editor - Widget Styles

| | |
|---|-----|
| Open a Theme in the Theme Editor | 113 |
| Creating a new Theme | 113 |
| Opening an Existing Theme | 115 |
| Theme Editor from the Project View | 115 |
| Setting the Current Theme from the Project View | 117 |
| Theme Editor Screen | 118 |
| Theme Editor Tab | 119 |
| Widget Category Tabs | 120 |
| Widget Properties in Theme Editor | 122 |
| 'Default.theme' in a New Project | 123 |
| Create a Widget Style for a Theme | 124 |
| Using Widget Styles | 127 |
| Default to Default Style: | 129 |

This sections describes theme and widget style editing:

Multiviewer video walls contain **Tiles** which contain various graphical widgets. Orbit MapView schematics are similar and also contain various graphical widgets. Widgets display information on-screen.

The appearance of a widgets across an Orbit project can be changed quickly with the use of **Themes** and **Styles**.

Alternative widget versions (styles) can be styled by editing stylistic widget properties (*non*-functional properties) and storing an alternative version of a widget (a **Style**) under a **Theme**. For example, audio bars on a video wall can be styled with left-handed and with right-handed widget versions, for use on-screen depending on their on-screen position.

Widgets styles are grouped under themes. Default widget styles are supplied in a new multiviewer Orbit project under a **Theme**.

Widget **Styles** and **Themes** can be viewed and edited with Orbit's **Theme Editor**. Viewing and editing these widget **Styles** and **Themes** requires a user to have editing privileges.

4.1 Open a Theme in the Theme Editor

The **Theme Editor** is used to manage Orbit **Themes** and widget **Styles**. It is accessed from the Orbit multiviewer **Project Home** screen:

4.1.1 Creating a new Theme

1. Click the **Themes** icon in a **Project Home** screen.
A list of theme designs appears below the **Themes** icon, see Figure 107.
(In a new project, there will be no themes and the list is empty.)

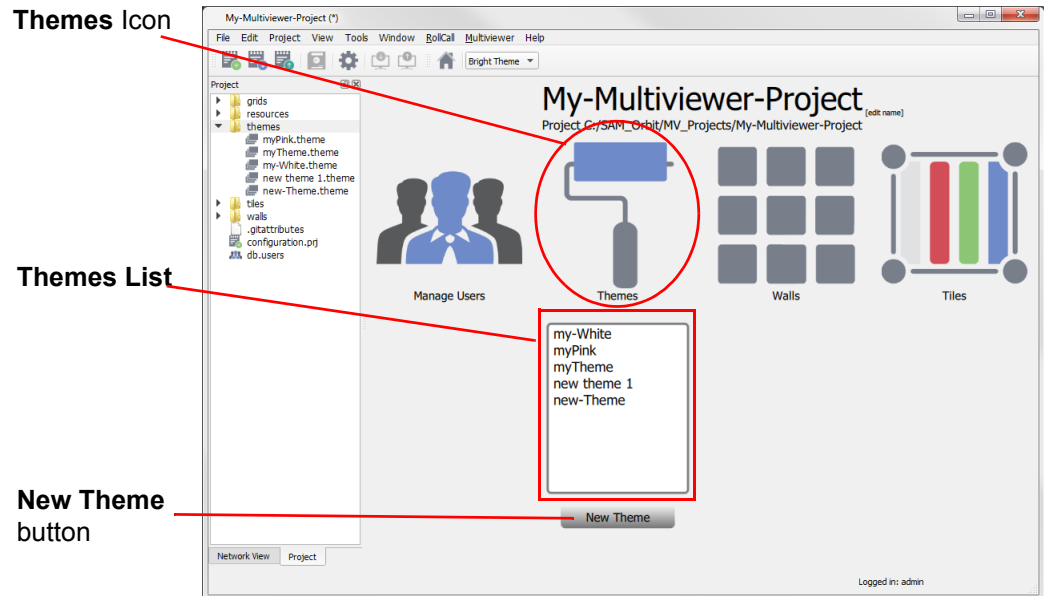


Figure 107 Orbit Project Screen - Themes Icon

2. Click the **New Theme** button, located below the **Themes List**.
A **New Theme** dialog appears. See Figure 108.

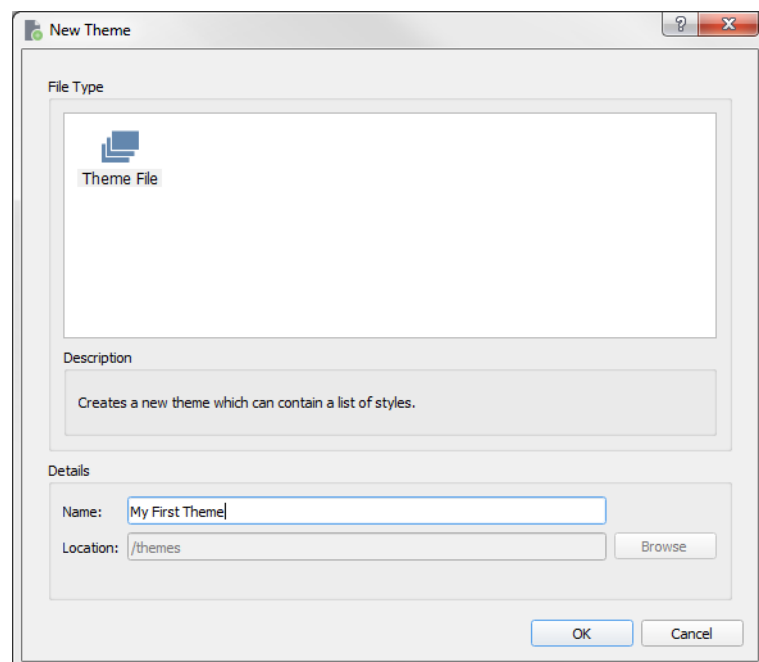


Figure 108 New Theme Dialog

3. Enter a name for the new **Theme** in the **Name** text box.

The default location of the new **Theme's** Orbit design file is the current project's '**themes**' directory, located in the Orbit project folder. This is shown in the **Location** text box.

It is recommended that you use this default location.

Note:

All information relating to the theme will be stored in a design file within the Orbit project. A project may contain more than one theme.

4. Click **OK**. (Or click **Cancel** to return to the Orbit project screen.)

A message box appears asking if you wish the project's currently-used **Theme** ('current **Theme**') to be set to this new **Theme**. See Figure 109.

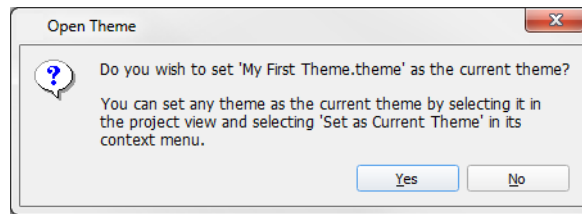


Figure 109 Open Theme Dialog

5. Click **Yes** to:

- Proceed to edit the new theme.
- And set it to be the current theme used by the project.

Click **No** to:

- Proceed to edit the new theme.
- And leave the current project theme unchanged.

6. The **Theme Editor** tab is displayed. See Figure 110.

A new **Theme** has been created and is open in the **Theme Editor**.

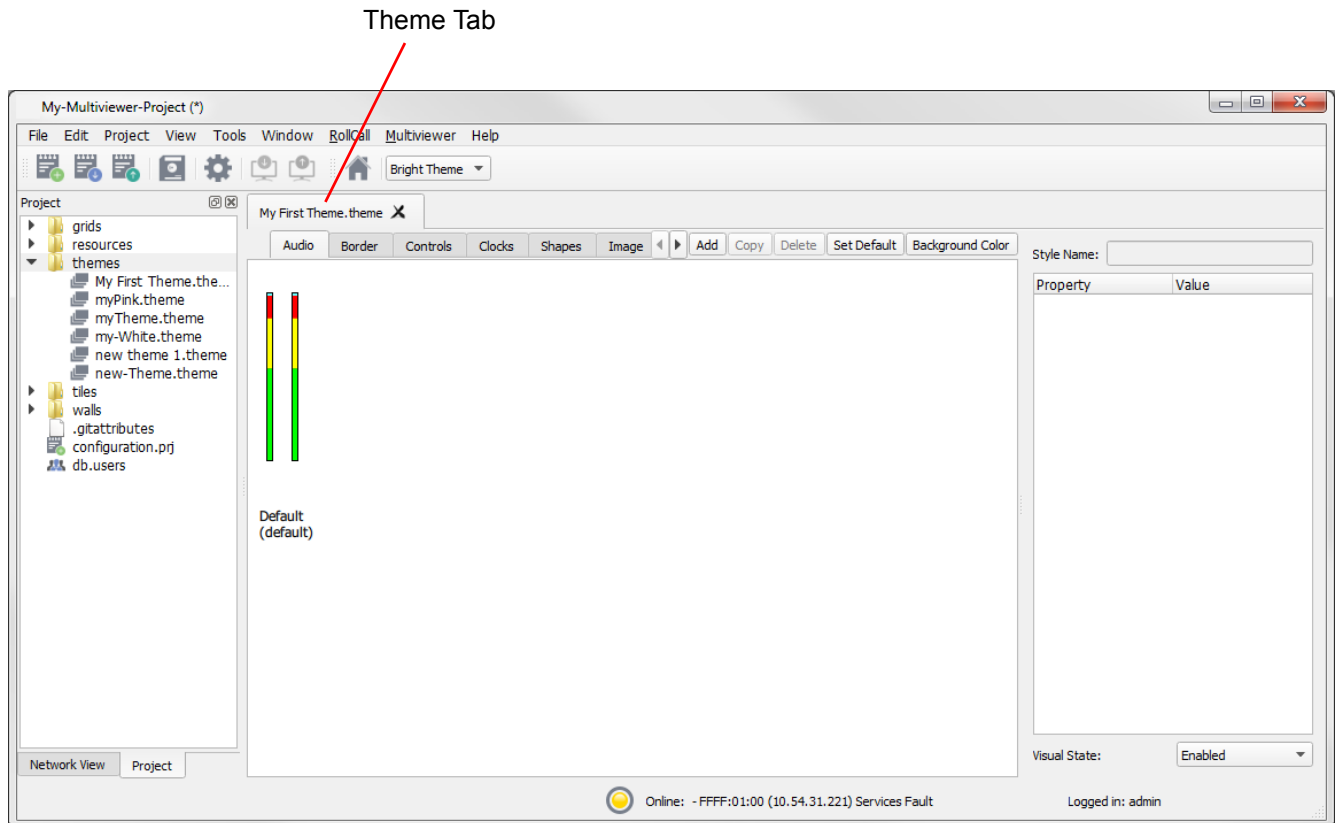


Figure 110 Theme Editor Screen

4.1.2 Opening an Existing Theme

1. In an Orbit project screen, click the **Themes** icon.
A list of existing themes appears below the **Themes** icon, see Figure 107 on page 113.
Click on a theme name in the **Themes List**.

Note: In a new project, there will be one “baseline” theme named 'Default'. This theme may be edited or a new theme created.

2. An **Open Theme** dialog is displayed (Figure 109 on page 114).
Click **Yes** or **No** as required.
3. The **Theme Editor** tab for the selected theme is displayed.

The existing theme is now opened in the **Theme Editor**.

4.1.3 Theme Editor from the Project View

The **Project View** may be used to access the **Theme Editor** via a pop up menu in the **Project View**. Using this, a theme may be:

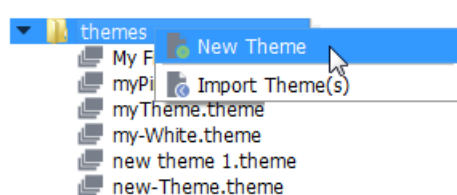
- Created (added).
- Imported from another Orbit Project.
- Renamed.
- Copied.
- Deleted.
- Set as the Current Theme in use on the project.

To create a new theme:

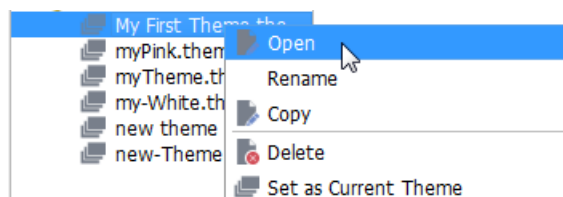
1. In the **Project View**, right-click on the 'themes' sub-folder name.
2. Click **New Theme** in the pop up menu. See Figure 111a.

To open an existing theme:

1. In the Project View, expand the 'themes' sub-folder to show the project's existing themes.
2. Right-click on the theme name required. Click **Open** in the pop up menu. See Figure 111b.



a) New Theme

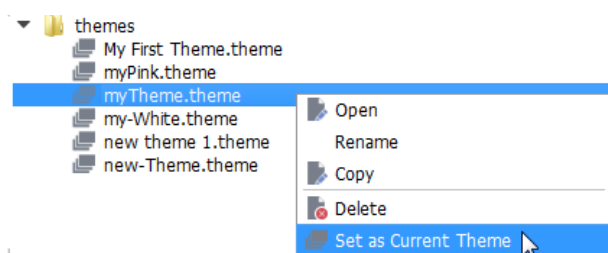


b) Open Theme

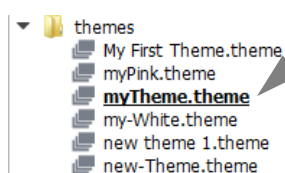
Figure 111 Project View: a) New Theme, b) Open Theme

4.1.4 Setting the Current Theme from the Project View

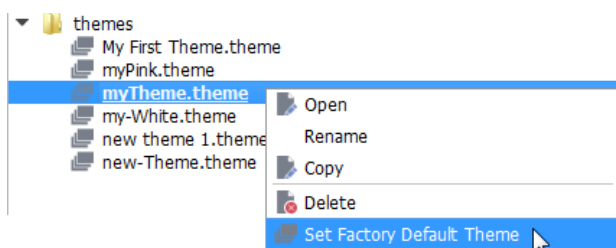
The **Project View** may be used to access the **Theme Editor** via a pop up menu in the



- a) Right-click and select 'Set as Current Theme' to set 'myTheme' as the Current Theme



- b) 'myTheme' shown underlined, indicating it is the Current Theme.



- c) Right-click now shows 'Set Factory Default Theme' menu item

Figure 112 Project View - Setting the Project's Current Theme

4.2 Theme Editor Screen

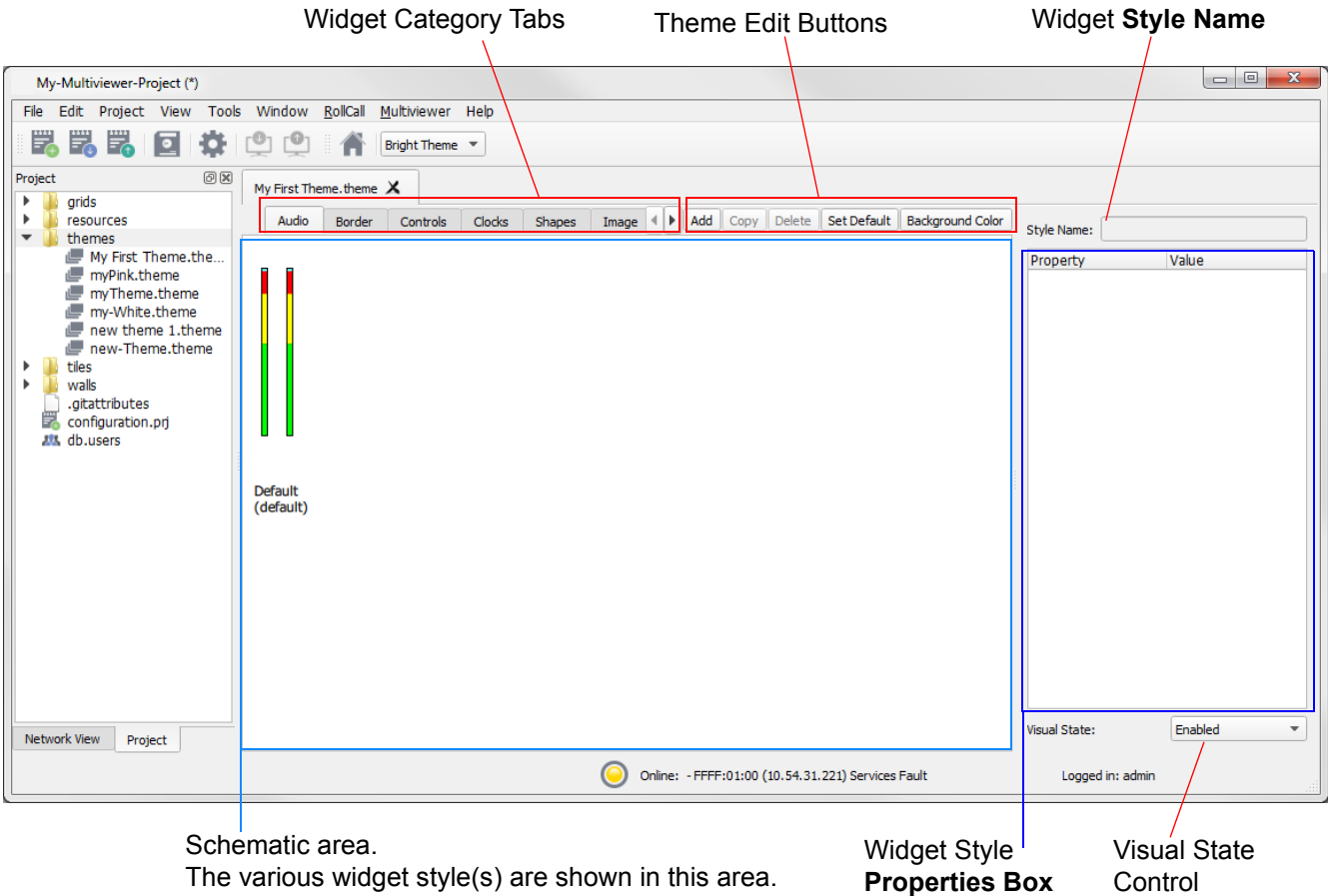


Figure 113 Theme Editor Tab

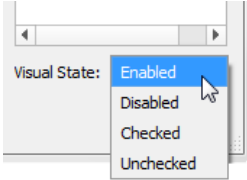
| Item | Description |
|--|---|
| Theme Edit buttons: | |
| Add (Style) | Button. Click to add a new widget style. |
| Delete (Style) | Button. Click to delete selected widget style. |
| Set Default (Style) | Button. Click to set the selected widget style as the default widget style to use for the Theme. |
| Background Color | Button. Click to set the Theme Editor schematic area background color. |
| Other items: | |
| Style Name | Text box. Enter a name for the widget style. |
| Visual State | Drop down box. Defines on-screen state of a control widget (e.g. check box or button): |
|  | |

Table 21 Theme Editor Buttons and Controls

4.2.1 Theme Editor Tab

The **Theme Editor** tab is labeled with the **Theme** name and a “.theme” suffix, for example, see Figure 114. A file of this name appears in the Orbit project and is shown in the **Project View** in the ‘themes’ sub-folder.

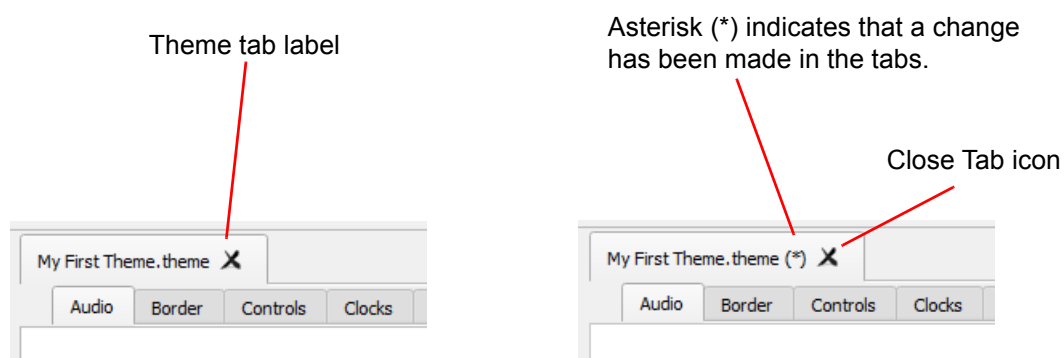


Figure 114 Theme Tab Label, (*) and Close Tab Icon

If a change has been made to settings in **Theme Editor** screen, an asterisk (*) is appended to the **Theme** tab label, see Figure 114.

Save changes using the **Save File** icon in the Orbit screen’s main tool bar.

Click the **Close Tab** icon, located next to the tab label, to close the **Theme** tab. If there are any unsaved changes in the tab, a prompt appears to ask to save any changes.

4.3 Widget Category Tabs

The **Theme Editor** screen contains a tab for each category of widget. All the **Widget Category Tabs** are shown in Figure 115.

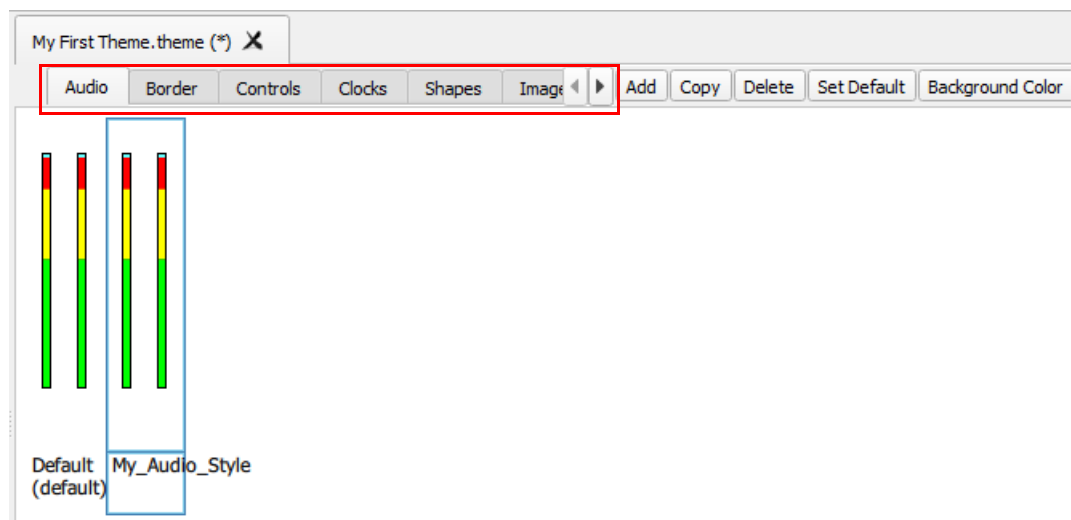


Figure 115 Widget Category Tabs

Within each category, if there is more than one type of widget in the category, then **Widget Side-tabs** are displayed, one side-tab per widget type. See Figure 116.

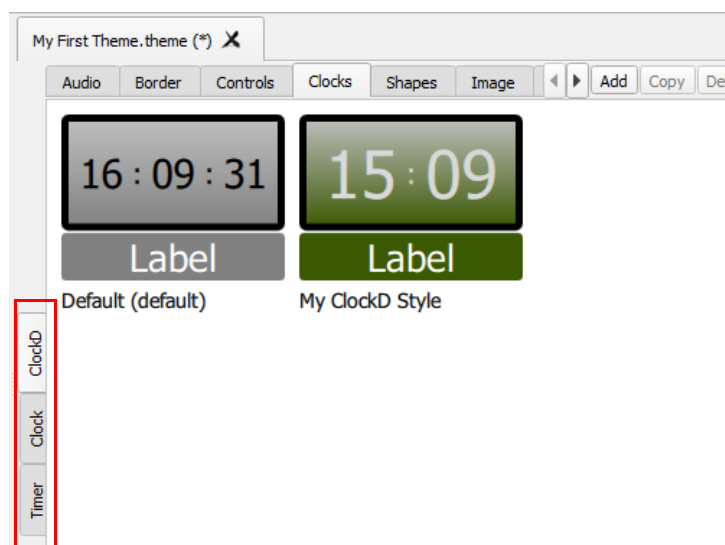


Figure 116 Widget Side-Tabs

If the Orbit **Theme Editor** schematic area is too small, the **Theme Editor** schematic area shows horizontal and vertical tab scroll controls to allow access to the top and side tabs. See Figure 117.

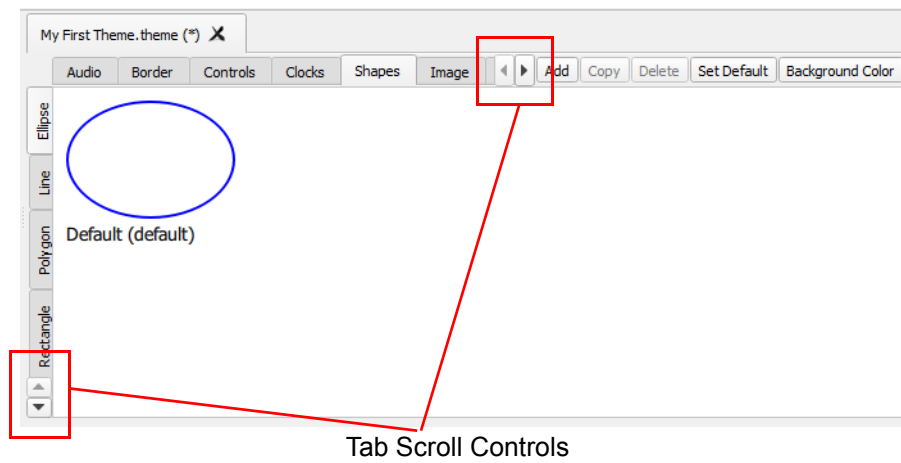


Figure 117 Theme Editor Scroll Bars

Each Widget Tab/Side-tab shows one or more widget styles for the **Theme** being edited.

4.4 Widget Properties in Theme Editor

Click on a **Widget** style to select it and the editable properties are shown in the **Properties Box**. See Figure 118.

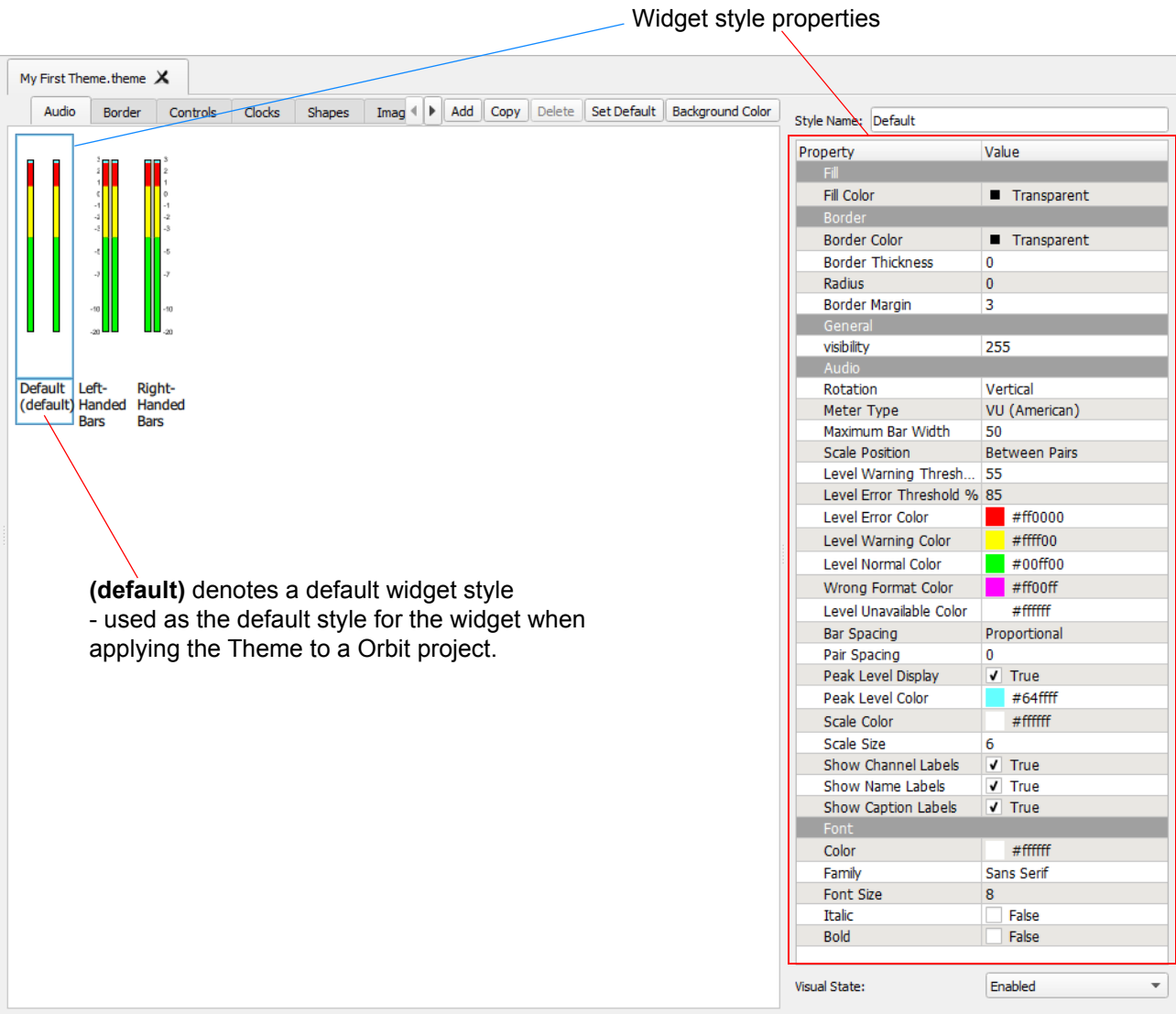


Figure 118 Widget Style Properties

Note:

(default)
The default widget style used by the **Theme** being edited when adding a widget to a **Tile** or **Schematic** is denoted by “(default)” after the widget style name.

4.5 'Default.theme' in a New Project

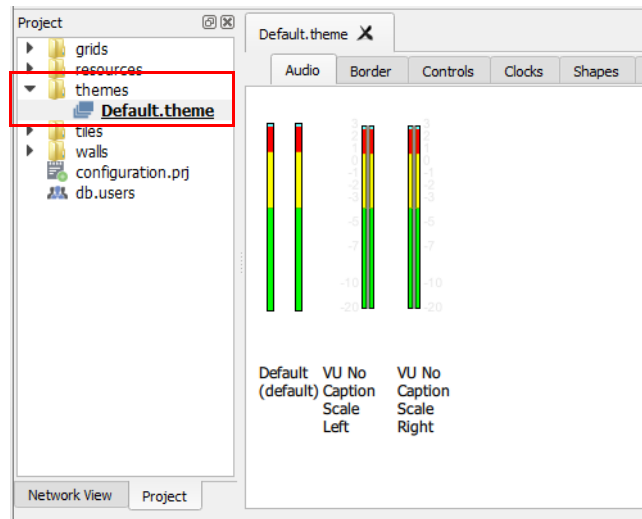


Figure 119 Default.theme

Orbit widget styles are supplied for each widget type in the **Default Theme** (default.theme' in the **Project View**).

Other widget styles may be created and added to the **Default Theme** or added to a new **Theme**. A new **Theme** may be created from another **Theme**. Thus a new **Theme** can be created based on the **Default Theme**.

Note:

Default Theme and the default widget style (default):

The **Default Theme** is part of a new Orbit project.

Each widget in a **Theme** has a default widget style. This is denoted by “**(default)**” after the widget style name, shown in the **Theme Editor**.

This default widget style is used by default when adding a widget to a **Tile** when a widget style is not present in the applied **Theme**.

4.6 Create a Widget Style for a Theme

This example creates two extra alternative Audio widget styles - a left-handed and a right-handed version of the audio bars widget:

1. Create a new **Theme**.
The **Theme Editor** tab for the new theme is displayed and a new 'themes' item appears in the **Project View**.
2. Click the **Background Color** button and change the **Theme Editor** background color to a mid-gray.
(This means it will be easier to see the extent of the widgets.)
See Figure 120.

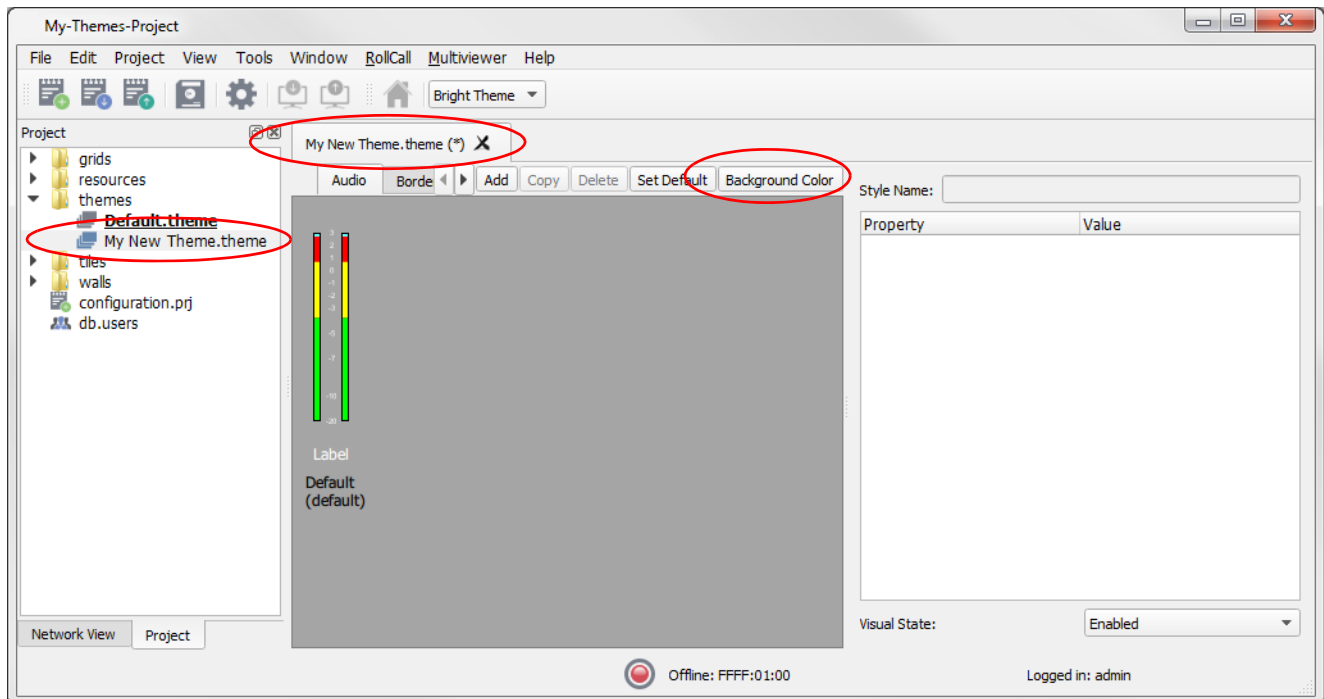


Figure 120 New Theme Tab

3. Click **Add**.
A new widget style graphic image is shown.
Click on the new style image to see its properties. See Figure 121.

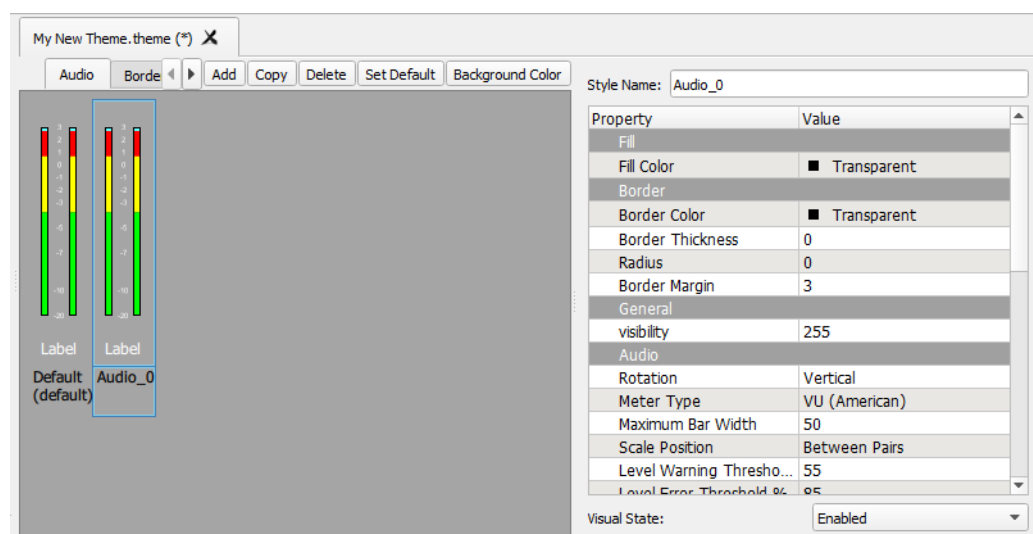


Figure 121 New Widget Style

4. Enter a name for the widget style in the **Style Name** text box.
For example, 'My Left-Handed Bars'.
5. Modify the **Audio > Scale Position** property value to **Right** (Right)
6. Modify the style properties as required for this style (Color etc). For example, see Figure 122.

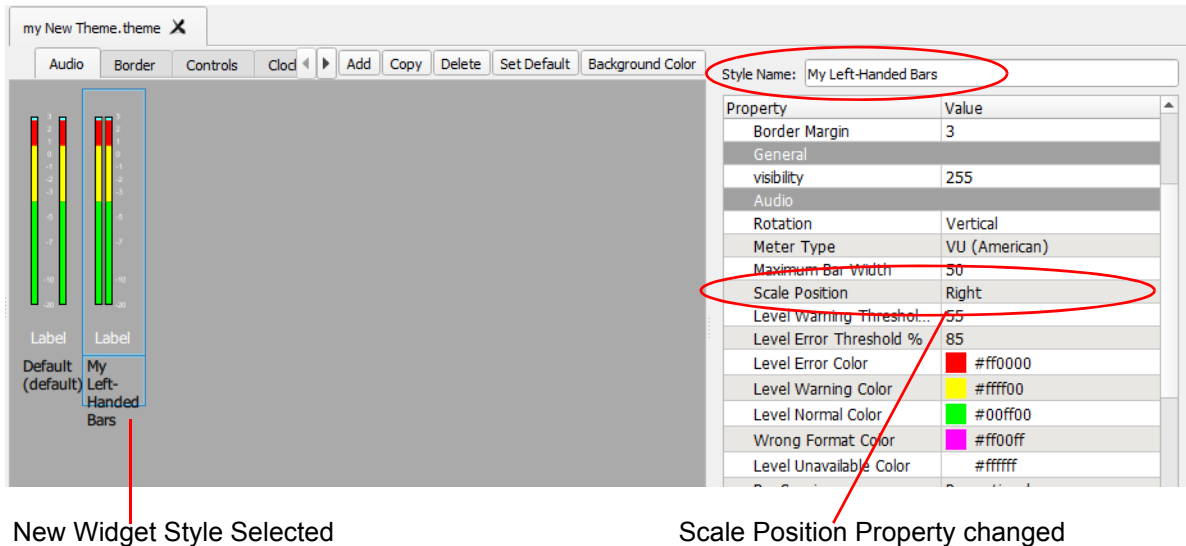


Figure 122 New Widget Style with modified Properties and Name

7. Click **Add**.
A second new widget style graphic image is shown.
8. Click on the new style image to see its properties.
Enter a name for the second new widget style in the **Style Name** text box.
Modify the style properties as required for this style.
For example, see Figure 123.

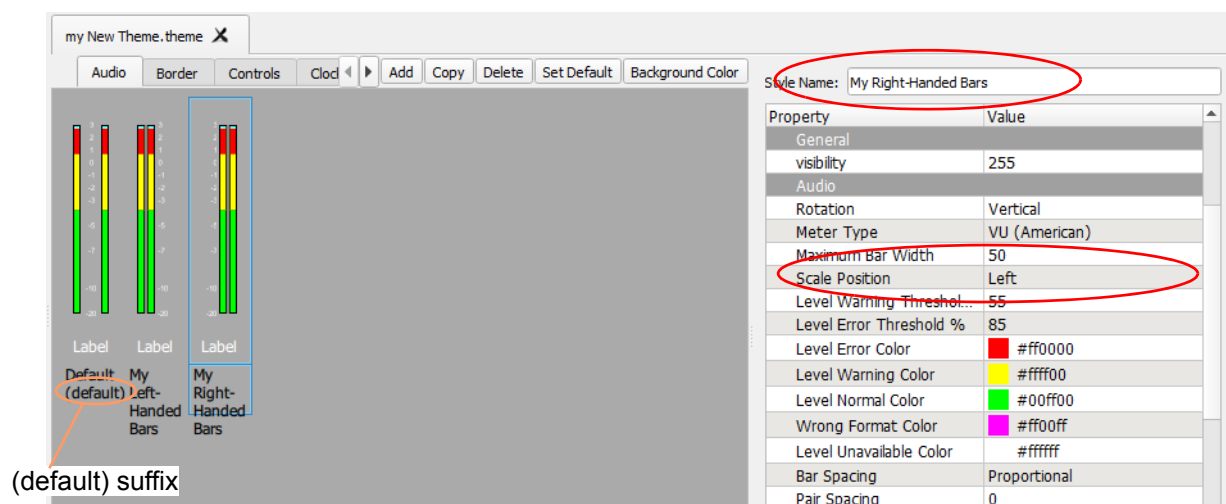


Figure 123 Second New Widget Style

9. Note the **“(default)”** suffix adjacent to one of the widget style names in the schematic area. See Figure 123. This indicates that this version of the widget style will be used, by default, when the widget is dragged onto a tile.
10. Click the **Save File** icon in the main tool bar.
This saves the **Theme** and the widget **Styles** it contains.

This example has now created an extra two alternative Audio widget styles - a left-handed and a right-handed version.

When the theme is applied to a project, these new widget styles are all available when designing tiles in the Tile Editor.

To set the default widget style for a theme:

If a theme does not contain the style for a widget, then the default style for that theme is used.

To set which widget style is used as default for a widget for a theme:

1. Open the Theme in the **Theme Editor**.
2. Select the style in the **Theme Editor**.
3. Click **Set default**.
“(default)” is appended to the style name in the **Theme Editor**.

To delete a style:

1. Check that the style to be deleted is not set as ‘default’.
If it is set as ‘default’, then set another style as default before proceeding to delete.
2. Select the style in the **Theme Editor**. Click **Delete**.
The style is removed.

4.7 Using Widget Styles

This example uses a **Theme** which contains two alternative **Audio** widget **Styles**, 'My Left-Handed Bars' and 'My Right-Handed Bars'.
The **Theme** is set as the current **Theme**.

1. Set the current theme to be the **Theme** with the **Audio** widget alternative **Styles** in it. (**Project View**, right-click on theme name, select 'Set as Current Theme'.)
2. Create a new **Wall** and fill it with video **Auto-Tiles**. See Figure 124.

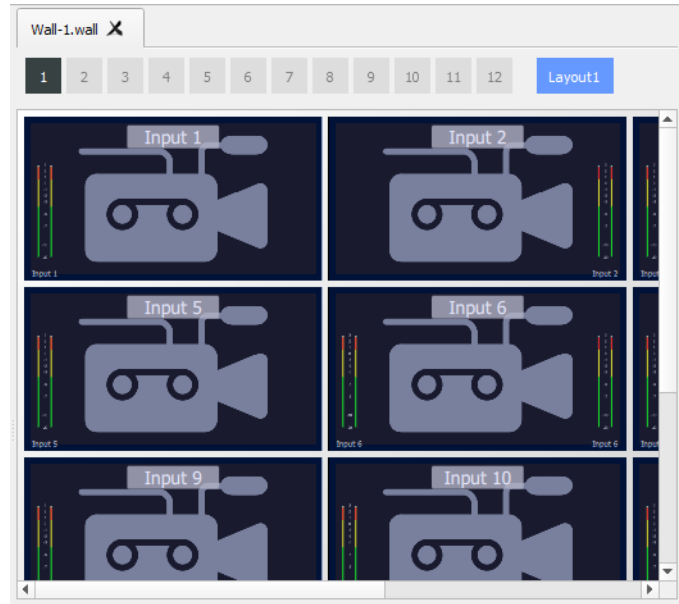
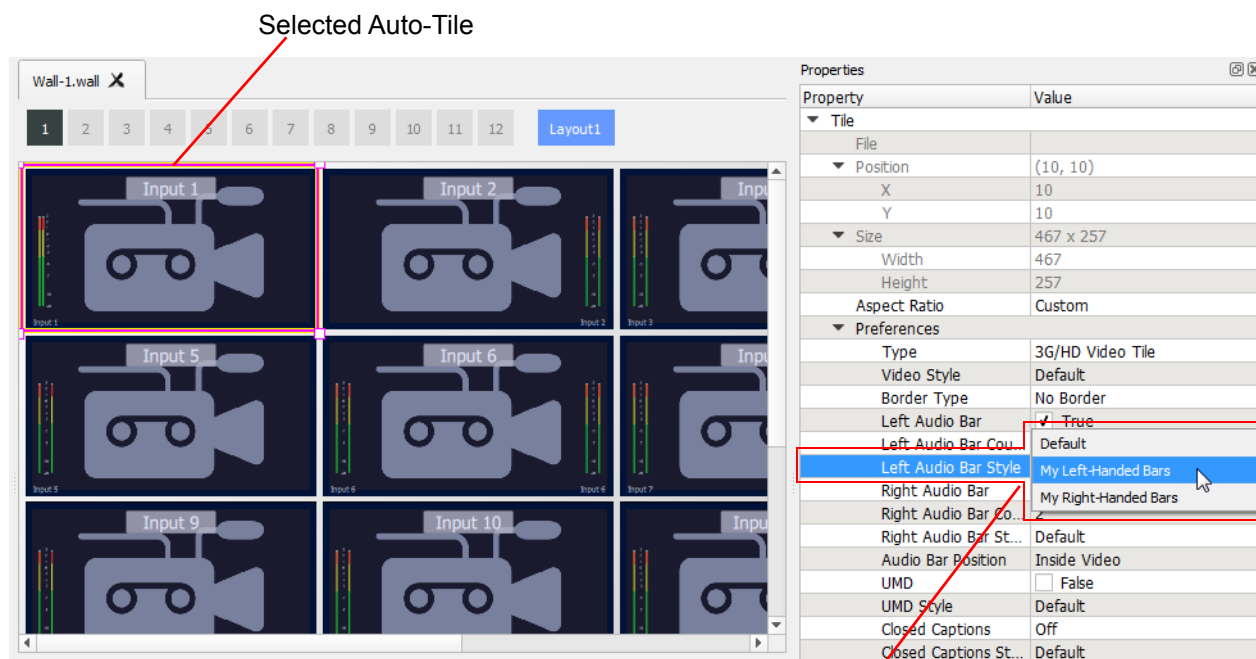


Figure 124 Video Auto-Tiles on a Wall

3. Select an **Auto-Tile** in the wall and look at its properties.

The **Audio** widget **Styles** available in the current **Theme** appear as possible values in a property of the **Auto-Tile**. The property value can be set at wall-level:

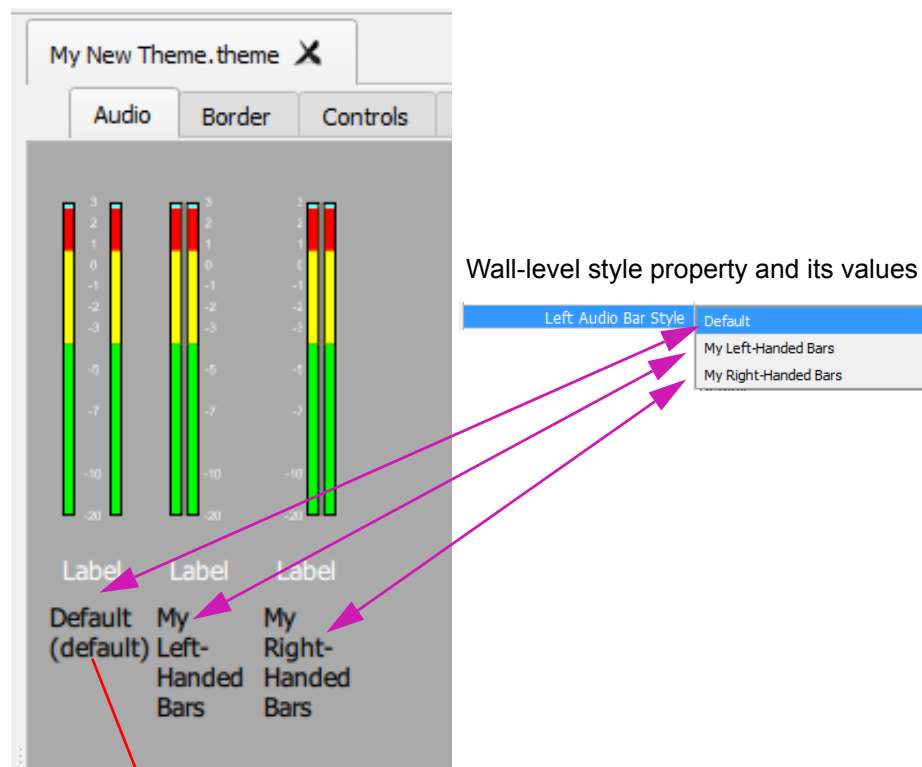
4. Set the 'Left Audio Bar Style' property to a left-handed audio bars style. See Figure 125 and Figure 126.



Auto-Tile's Audio Widget Style Property at Wall Level

Figure 125 Wall-Level Audio Widget Style Property

Audio widget styles in Current Theme



Wall-level style property and its values

Set to be the 'default' style to use for Audio bars

Figure 126 Widget Style Property at Wall Level

Continuing with this example, different video tiles can be variously set to have left- and right-handed styles with 2 or more bars. See Figure 127.



Different styles of (or number of) Audio Bars shown on tiles.

Figure 127 Audio Widgets Variously-Styled at Wall Level

4.7.1 Default to Default Style:

The Audio widgets on the **Auto-Tile** in Figure 127 now have their styles set to user-defined styles: 'My Left-Handed Bars' and 'My Right-Handed Bars'. The style properties may be set to these user-defined style names because the styles belong to the current **Theme**.

If the project's current **Theme** is now changed, however, then Orbit will try and find the same widget style names in the new current theme. If these names are *not* present in the new current theme, then the widget styles default to becoming the current theme's 'default' audio widget style.

5

5 Widget Behaviours and Bindings

Widget Behaviours and Bindings

| | |
|--|-----|
| Introduction | 131 |
| Behaviour and Binding Example Use-Cases | 131 |
| Behaviours and Bindings Graphical Editor | 134 |
| Behaviours and Bindings Editor Screen | 135 |
| Behaviours and Bindings Vertical Tabs | 137 |
| Behaviour and Bindings Editor - Tool Bar | 138 |
| Behaviour and Bindings Editor - Block Diagram Area | 138 |
| Widget Property Types | 142 |
| Using Orbit Variables with Strings | 143 |
| Enumerations (Named Value) | 143 |
| Setting Color Values | 143 |
| Video Auto-Tiles | 144 |

5.1 Introduction

Orbit can be used to design:

- Multiviewer video walls, built from graphical tiles.
- Graphical control /monitoring screens, comprising graphical schematics.

A Orbit **Tile** or **Schematic** are designed in a similar way: Their on-screen appearance is composed of graphical widgets and their on-screen functionality is constructed with **Behaviours** and **Bindings**, which define how the widgets interact and behave.

Orbit offers some built-in (multiviewer) tiles (**Auto-Tiles**), which are automatically-generated, and which have fixed, built-in functionality. These cover the majority of tile use-cases. For other use-cases, a **Custom Tile** is used and tile functionality may be defined and modified by the user by editing **Behaviours** and **Bindings** for the tile and widgets.

Note:

Creating Custom Tiles:

It is recommended that a **Custom Tile** is created by using an **Auto-Tile** as a starting point. Because **Auto-Tiles** already have some built-in functionality set up with **Behaviours** and **Bindings**.

Note:

Auto-Tile Functionality and Tile Editing:

An **Auto-Tile's** various functions cannot be changed.

When attempting to edit an **Auto-Tile**, the tile is first converted into a **Custom Tile** before editing. The user is prompted when this will occur.

Widgets are graphical items on **Tiles**, or **Schematics**, with properties that change the widget's appearance, for example, font size, flashing border, text label, color or transparency. The behavior of a widget is how its appearance changes according to various conditions; this is defined with Orbit **Behaviours** and **Bindings**. In this way, widget properties may be adjusted according to one or more system status data values, states or settings etc.

A **Behaviour** gets, puts or holds data. It can get information from or pass information to the outside world, to the multiviewer device itself or to a widget.

For example, getting status information from a piece of video equipment or controlling a GPI output on a multiviewer.

A **Binding** processes data. It connects a **Behaviour** to a widget or to another **Behaviour**. Typically, it binds an output of a **Behaviour** to a widget property.

Behaviours and **Bindings** are listed in C "List of Behaviours and Bindings" on page 332.

For example, asserting a widget property color value based on whether a status message indicates an 'error' or a 'warning'.

5.1.1 Behaviour and Binding Example Use-Cases

A **Behaviour** and **Binding** together can implement some functionality on a Orbit **Tile** or **Schematic**. A widget may be bound to more than one **Binding** where several widget properties may need controlling for some desired behavior.

For example, LED color, border color, border visibility and border flashing properties.

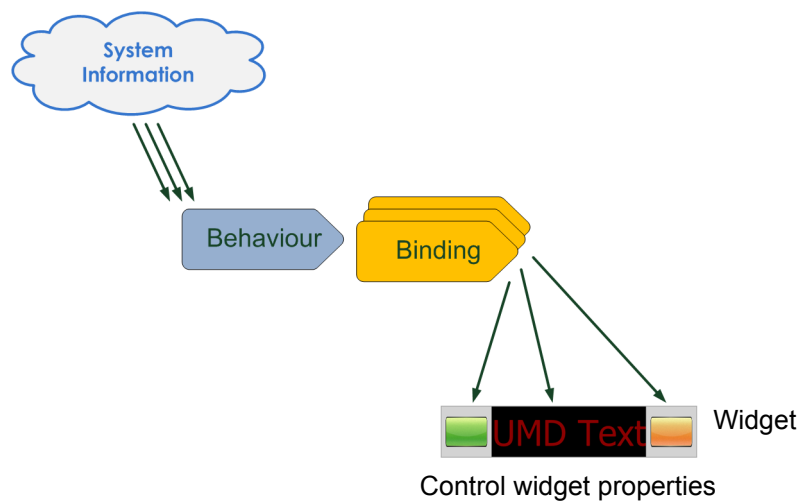


Figure 128 Behaviours and Bindings

Example **Behaviour** and **Binding** use-cases:

- On-screen status:
A **Behaviour** can obtain status information from some device and a **Binding** can connect to an on-screen LED indicator or to the showing of some (meaningful) graphic.
- TSL status on-screen with a UMD widget:
A **Behaviour** listens to TSL messages in the system and a **Binding** passes information to a UMD widget's lamps and text display.
- Bespoke TSL status on-screen:
A **Behaviour** listens to TSL messages in the system and a **Binding** connects the resulting information to various on-screen widgets. For example, an LED, an image, a colored background and a text field.

Figure 129 shows some on-screen results.

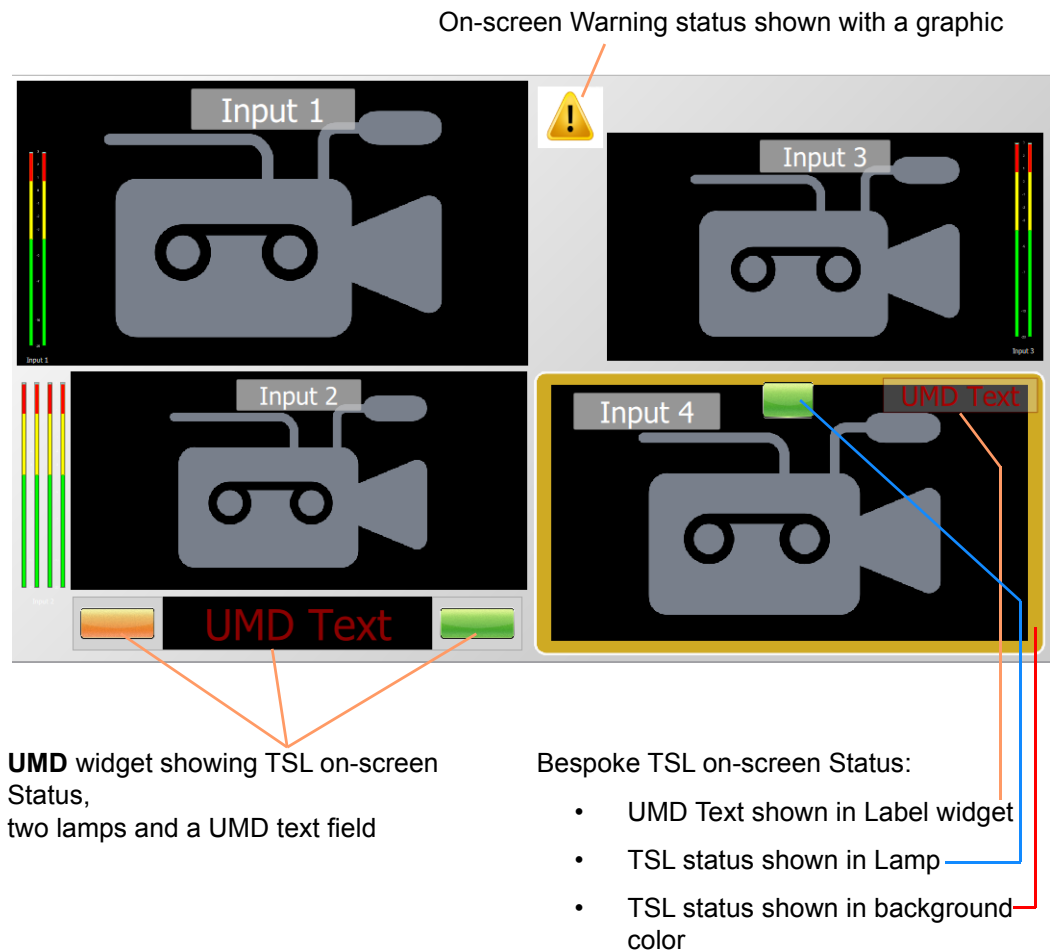


Figure 129 Example On-Screen Information from Behaviours and Bindings

After the widgets on a **Tile** or **Schematic** have been placed, **Tile/Schematic** functionality can be created and edited. For each placed widget:

- Instances of one or more **Behaviour** and **Binding** are made in a graphical editor.
- **Behaviour** and **Binding** properties are configured.

In Orbit, there is a range of various **Behaviours** and various **Bindings**.

Behaviours and **Bindings** access, process and pass on data. Certain data types are used by **Bindings** to link to widget properties.

Behaviours and **Bindings** may also be added to a **Tile** or to a **Schematic** itself, instead of to a widget. Typically, this would be used for **Behaviours** and **Bindings** that are common to more than one widget on the **Tile/Schematic**.

5.2 Behaviours and Bindings Graphical Editor

In the **Tile Editor**, click on the **Edit Behaviours** icon in the **Advanced Tool Bar** to access the graphical editor for **Behaviours** and **Bindings**, see Figure 130a.
The graphical **Behaviours and Bindings Editor** is shown in the lower part of the **Tile Editor**, see Figure 130b.

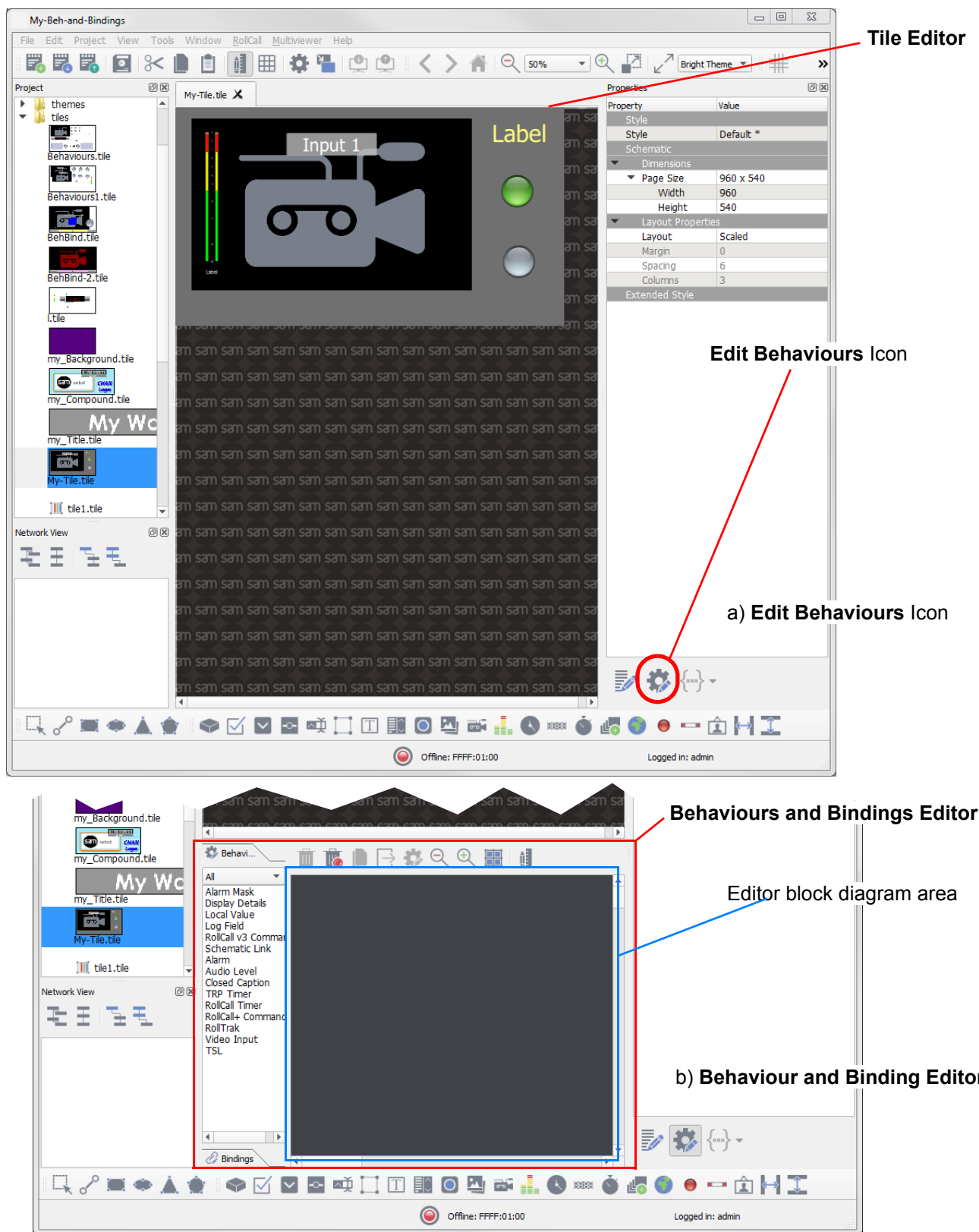


Figure 130 Behaviour and Bindings Graphical Editor

When a widget is selected on the **Tile**, any **Behaviours** and **Bindings** that are set up for that widget are shown in the block diagram area, see Figure 131.

If no widgets are selected, then the **Behaviours** and **Bindings** that are set up for the **Tile/Schematic** itself are shown.

If the selected item has no **Behaviours** nor **Bindings**, then the block diagram is empty.

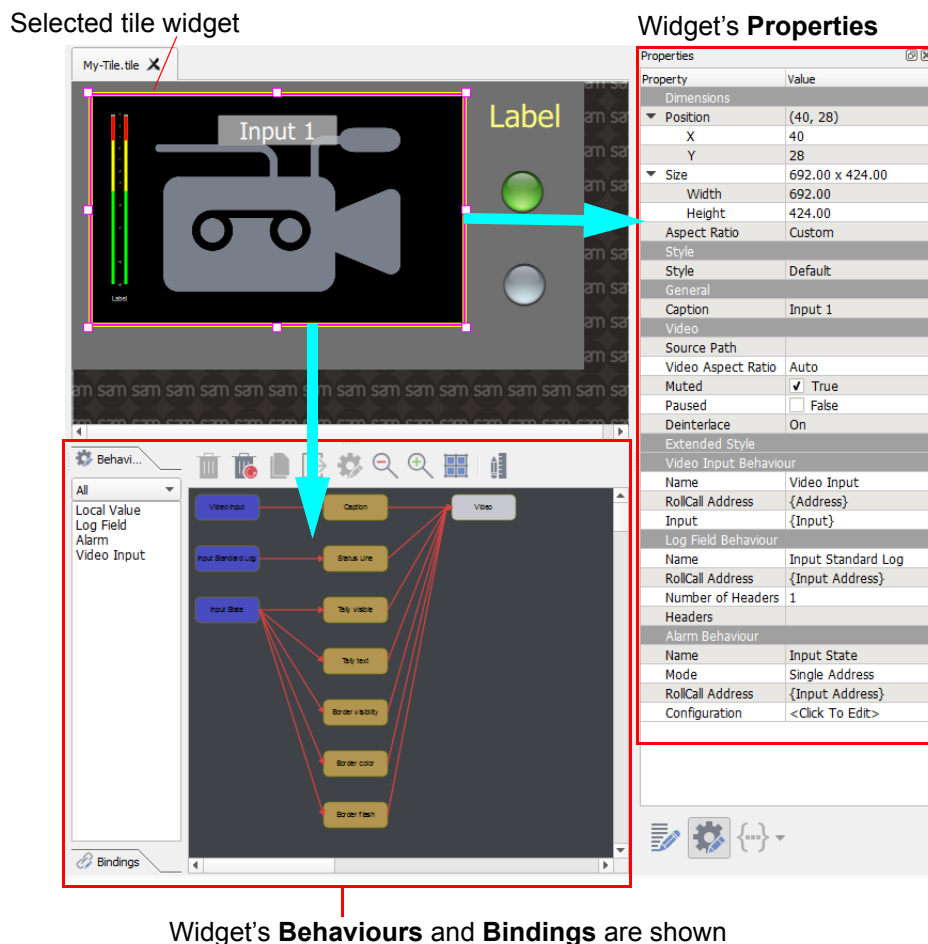


Figure 131 Video Widget Behaviours and Bindings

5.2.1 Behaviours and Bindings Editor Screen

The **Behaviours and Bindings Editor** screen is shown in Figure 132.

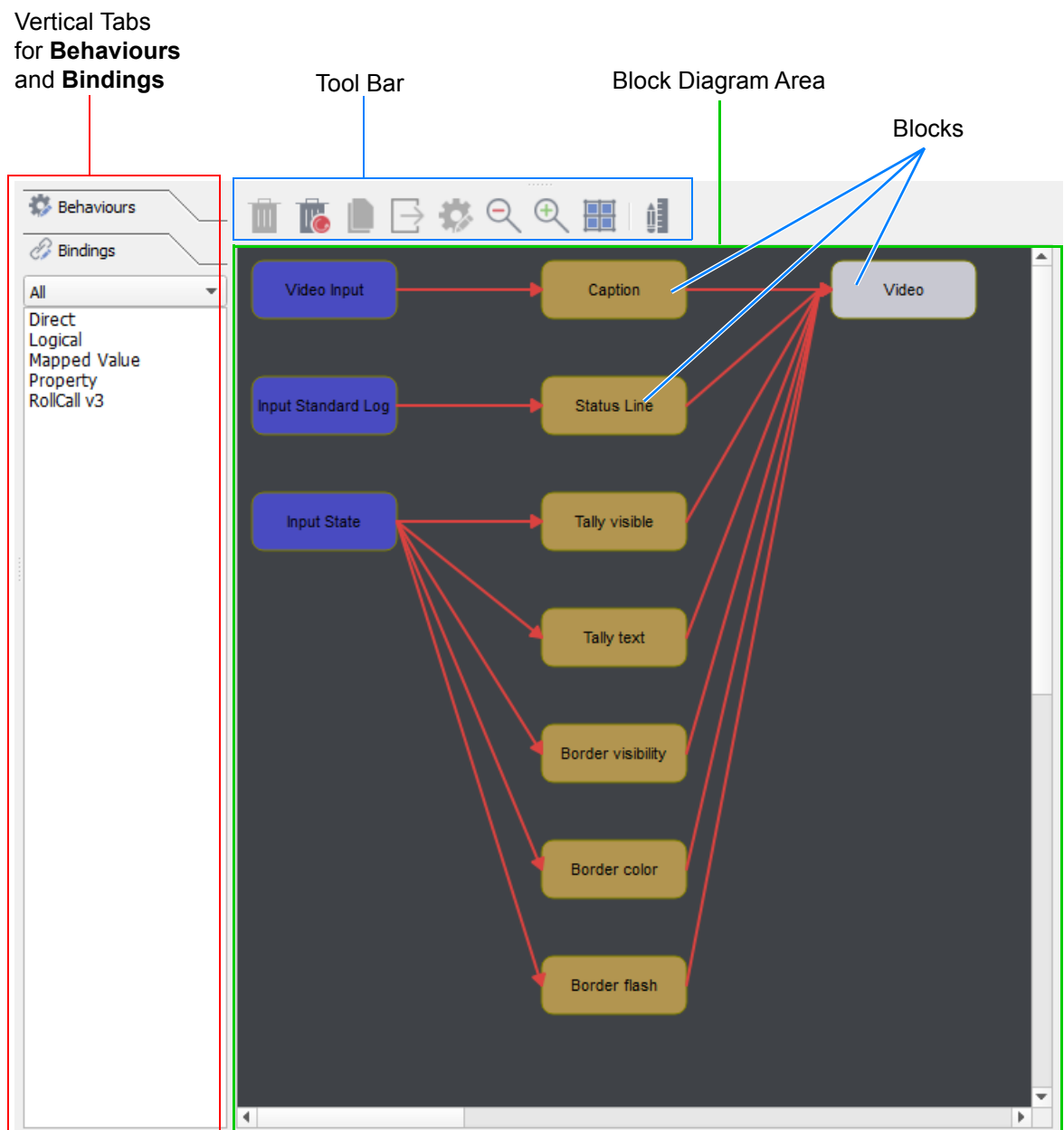


Figure 132 Behaviours and Bindings Editor Screen

5.2.2 Behaviours and Bindings Vertical Tabs

The available **Behaviours** and **Bindings** for the selected widget are listed; there is a tab for **Behaviours** (see Figure 133a) and a tab for **Bindings** (see Figure 133b). Each tab lists the respective available **Behaviours** or **Bindings**. The list may be filtered with the drop-down selection box. See Figure 133.

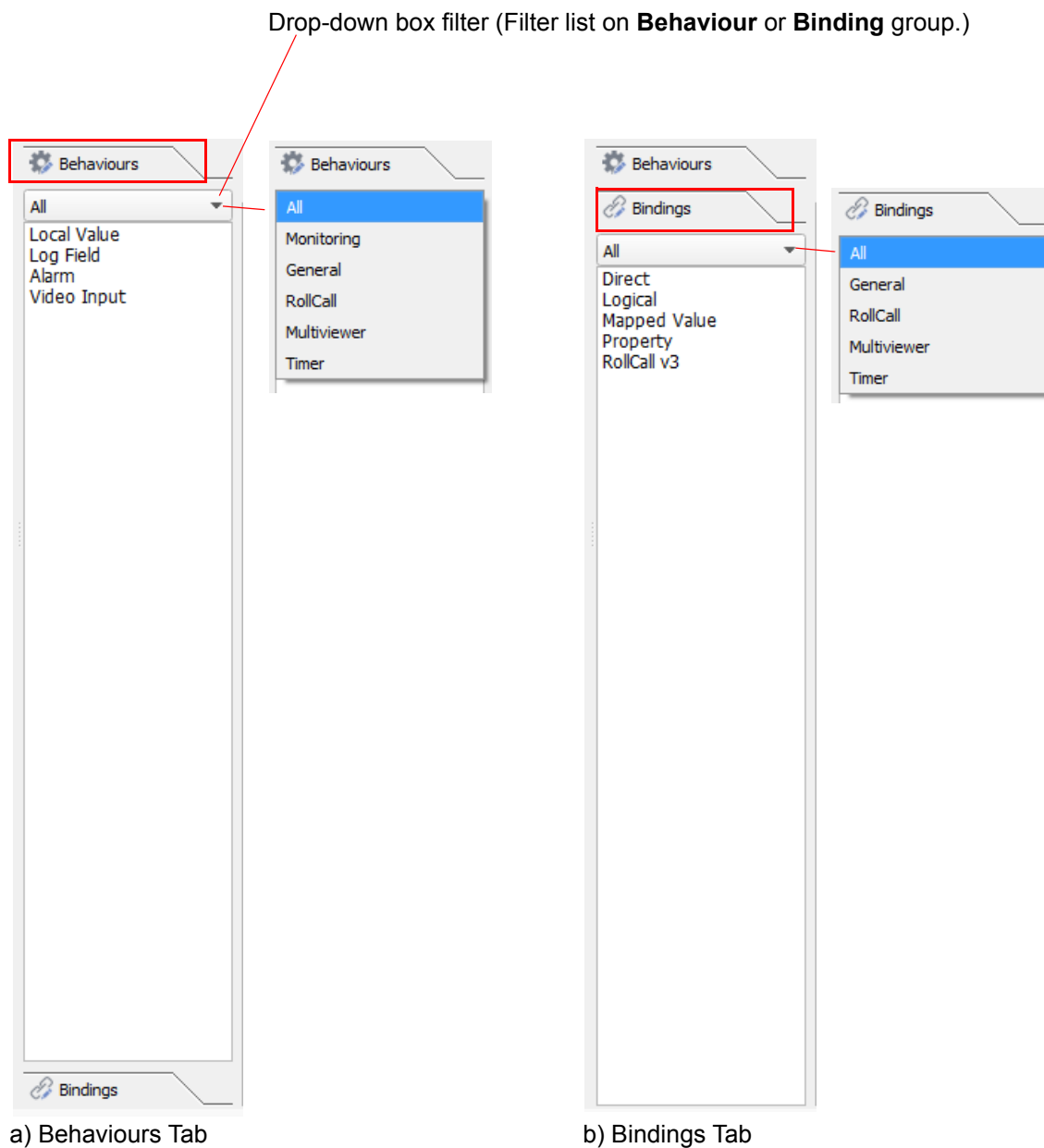


Figure 133 Behaviours and Bindings Tabs (Video Widget Example):

a) Behaviours Tab.

b) Bindings Tab.

The different **Behaviour** types are described in Section C.1 “List of Behaviours” on page 333.

The different **Binding** types are described in Section C.2 “List of Bindings” on page 352.

5.2.3 Behaviour and Bindings Editor - Tool Bar

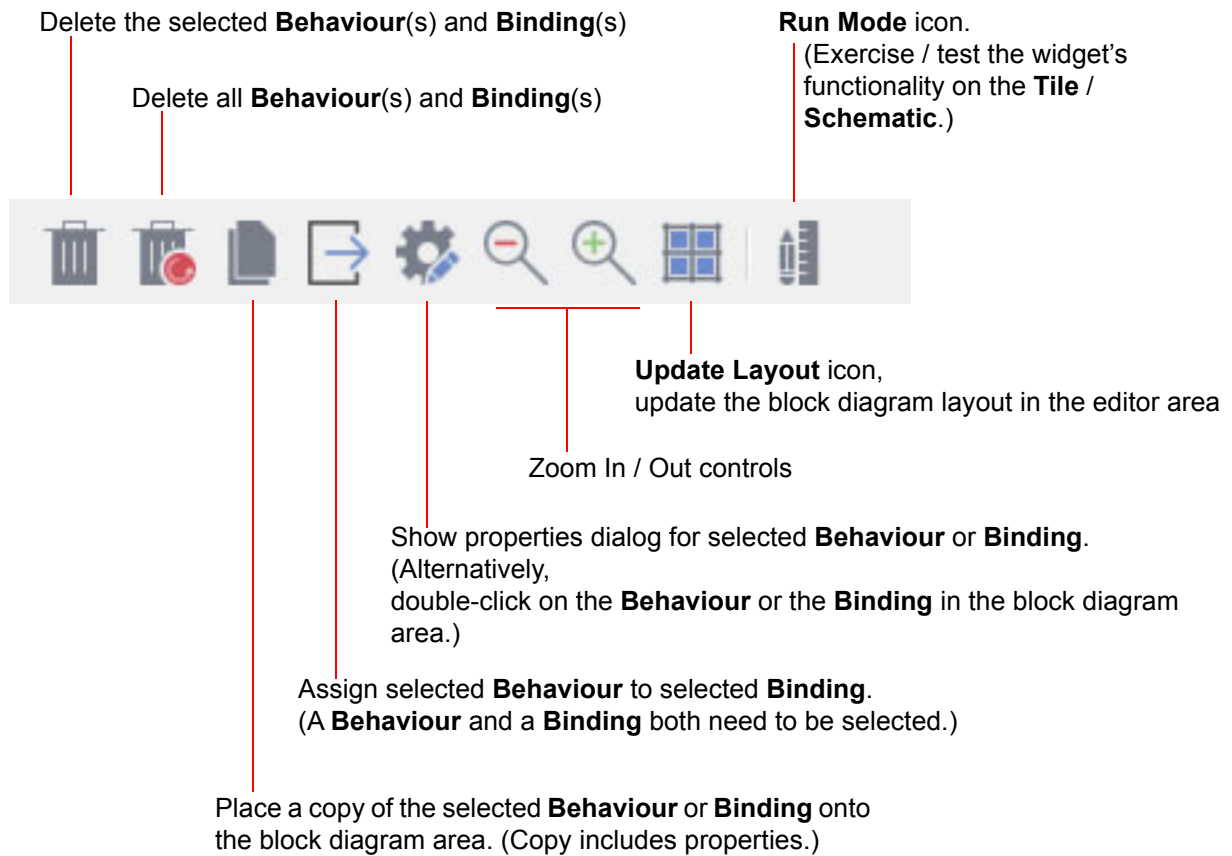


Figure 134 Behaviours and Bindings Editor Tool Bar

When a widget is being designed, the user can exercise the functionality of the **Tile/Schematic** by clicking the **Run Mode** icon. This toggles between Run Mode and Design Mode for the Tile. This does not require a license.

5.2.4 Behaviour and Bindings Editor - Block Diagram Area

When a widget is selected in the **Tile Editor**, the **Behaviour and Bindings Editor**'s block diagram area shows a block diagram representation of the **Behaviours** and **Bindings** for that widget. See example in Figure 135.

If no **Behaviours** nor **Bindings** have been set up, then the block diagram area is blank.

Block color indicates:

Behaviour block is *not* found on the current block diagram;
instead, the **Behaviour** block is on another block diagram.
(i.e. The block diagram for another widget on the current **Tile** or **Schematic**, or
for the **Tile** itself, or for a **Schematic** in the Orbit project.)

Block color indicates:

Behaviour block is on the current block diagram.



Figure 135 Video Widget Example of a Behaviour and Binding Block Diagram

The diagram shows the structure of the instances of **Behaviours** and **Bindings** set up for the selected widget (or **Tile** itself, or **Schematic** itself). Each **Behaviour** and each **Binding** instance is represented by a block on the diagram.

Connection arrows show the **Behaviour-Binding** data connections.

Connections can be made within a **Tile/Schematic**. Connections cannot be made off of a **Tile/Schematic**.

I.e. Connections can be made to any **Behaviour** belonging to:

- The current widget.
- Or the **Tile/Schematic** itself.
- Or any other widget on the **Tile/Schematic**.

When a block diagram for a widget is shown, one block on the block diagram represents the widget itself.

Hover the cursor over a block to see a tool tip message. See Figure 136.

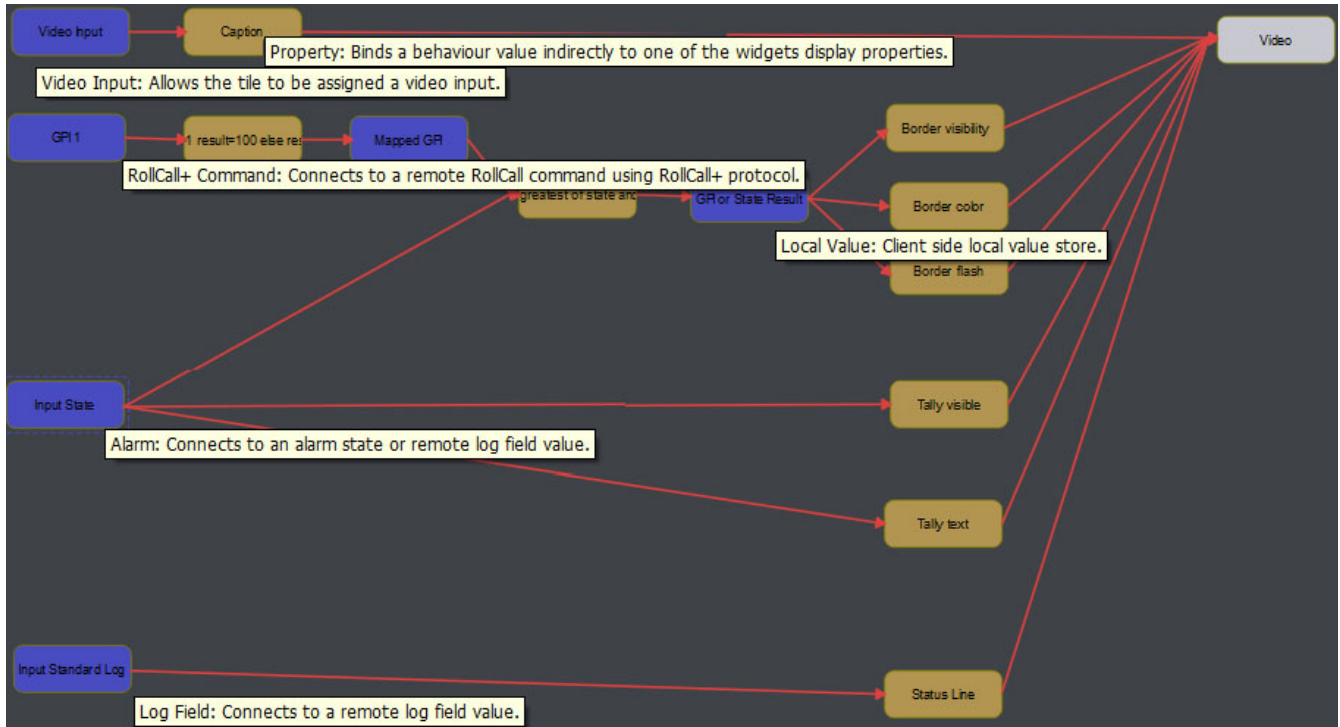


Figure 136 Tool Tip Examples

- Click on a **Behaviour** or a **Binding** block to select it.

- Double-click on a **Behaviour** or a **Binding** block to display its properties dialog. Example dialogs are shown in Figure 137.

Edit Audio Levels

| Property | Value |
|------------------|--------------|
| Name | Audio Levels |
| RollCall Address | {Address} |
| Input | {Input} |

Close

Edit Mapped Value

Common Properties

Name:

Source Behaviour:

Property to bind:

Bind Rules:

| Operator | Expression | Result |
|-----------------------|------------|--------------|
| Greater Than or Equal | 99 | ERROR (%1) |
| Greater Than or Equal | 49 | WARNING (%1) |
| Default | | OK (%1) |

Target: ☒ Widget ☐ Behaviour

Move Down Move Up Add Delete

Select

Figure 137 Example Behaviour and Bindings Properties Dialogs

Each **Behaviour** and **Binding** block can be given a name by the user. This helps to identify the block. A short descriptive name helps to document the block's function. For example, 'TSL 5.0', 'Caption Text', 'Fill Color' and 'Border Color'.

Orbit built-in variables can be used in the properties dialogs. For example, {Address} and {Input}.

Note:

Using system variables {Address} and {Index} in the **Behaviour** properties means that:

- The RollCall address of the Multiviewer is automatically used.
- The Video Input number assigned to the tile the widget is on is used.

This is more flexible than setting the properties to fixed values.

5.3 Widget Property Types

Widget properties may be set by **Bindings** and the main data types of a widget property are listed in Table 22.

| Data Type | Description | Widget Property Examples | Property Value Examples |
|-------------------------------------|--|---|--|
| String | A series of characters, treated as a group. | <i>RollCall Address</i> | <i>FFFF:02:01</i> |
| | Also see Section 5.3.1 “Using Orbit Variables with Strings” on page 143. | <i>Caption</i> | <i>Channel News</i> |
| Integer | An integer scalar value. | <i>Border Thickness</i> | <i>0</i> |
| | | <i>Font Size</i> | <i>12</i> |
| Real | Real number. | <i>Slider Position</i> | <i>0.732</i> <i>12.345</i> |
| Boolean | A value that is ‘true’ or ‘false’ (or ‘on’ or ‘off’; or ‘enabled’ or ‘disabled’; or ‘yes’ or ‘no’). Used for properties that are set with a check box. | <i>Caption Enable</i> <i>Background Visible</i> | <i>selected box (true),</i> <i>empty box (false)</i> |
| Enumeration (Enum) | A ‘named value’ data type. Properties that comprise a set list of items are implemented with an enumerated data type. The data type acts to select an item from the list. The data type uses a range of integer values each of which is given a name, an identifier. So, for our example, the following could be defined: 0 Left, 1 Right, 2 Center. Note: When setting enumeration type property values with a Binding , use an Integer type. See also Section 5.3.2 “Enumerations (Named Value)” on page 143. | <i>Alignment</i> | <i>Left</i> <i>Right</i> <i>Center</i> |
| Color | When entering color values into a text box, there are two methods supported: <ul style="list-style-type: none"> Hexadecimal: RGB(A) format. <ul style="list-style-type: none"> For example, ‘#FF00FF’ = magenta ‘#80FF0000’ = semi-transparent red SVG 1.0 color names: <ul style="list-style-type: none"> For example, ‘limegreen’. See also Section 5.3.3 “Setting Color Values” on page 143. | <i>Background Color</i> <i>Border Color</i> <i>Fill Color</i> | <i>#0000FF</i> <i>#800000FF</i> <i>#FFFFFFF80</i> <i>‘blue’</i> <i>‘darkblue’</i> <i>‘aquamarine’</i> |

Table 22 Widget Property Data Types

5.3.1 Using Orbit Variables with Strings

String widget property values can be replaced by Orbit variables: A variable name is placed inside curly braces. The value of the variable is used for the property value.

For example, a Orbit variable “Header” may be used in a widget property field thus:

```
{Header}
```

Note:

The variable “Header” can be specified either:

- At ‘project-level’ (**Project > Edit Variables**)
- or
- At ‘tile-level’ (deselect all widgets in the tile, then **Right-Click > Variables**)

5.3.2 Enumerations (Named Value)

For a list of items, a ‘named value’ (enumerated) variable can be used to select items from the list. The data type is a range of integer values where each list item is associated with an integer value of the data type.

A ‘named value’ data type uses an integer data type in a special way:

- A range of integer values are used.
- Values in the range are assigned a name, typically describing what it represents.

For example, for the ‘named value’ property ‘Alignment’:

0 means ‘Left’

1 means ‘Center’

2 means ‘Right’

5.3.3 Setting Color Values

When setting a color property through a **Binding**, there are two ways to express the color:

1. Color name:

A color name may be entered.

For example,

blue, lime, darkblue, aquamarine

The color names used are SVG 1.0 color names, defined at

<http://www.w3.org/TR/2003/REC-SVG11-20030114/types.html#ColorKeywords>

2. Hexadecimal:

A hexadecimal value may be entered, representing Red, Green, Blue and, optionally, a transparency value, i.e. (Alpha)RGB. If the transparency value, Alpha, is omitted, a solid, opaque color is defined.

For example,

```
#FF0000    for solid, opaque red.
#00FF00    for solid lime green (i.e. 'lime').
#0000FF    for solid blue.

#800000FF  for semi-transparent blue.
#FF0000FF  for solid, opaque blue.
#000000FF  for clear, totally transparent (blue).
```

5.4 Video Auto-Tiles

On a tile, the functionality of it and its widgets is defined with **Behaviours** and **Bindings**.

Auto-Tiles are provided for multiviewers with a framework of **Behaviours** and **Bindings** already set up. Some changes to **Auto-Tile** functionality can be made by editing widget properties.

For example, a video **Auto-Tile** includes an Audio Bars widget and a user can select the number of audio bars displayed by changing a property.

Thus, **Auto-Tile** functionality can be tailored.

If more extensive tile/widget functionality is required, then a **Custom Tile** is required and the framework of **Behaviours** and **Bindings** are edited in the **Tile Editor**.

An **Auto-Tile** may be used as the basis of a **Custom Tile**.

In this section, the inherent **Behaviours** and **Bindings** settings of a Video **Auto-Tile** are shown. Figure 138 shows a Video **Auto-Tile**.

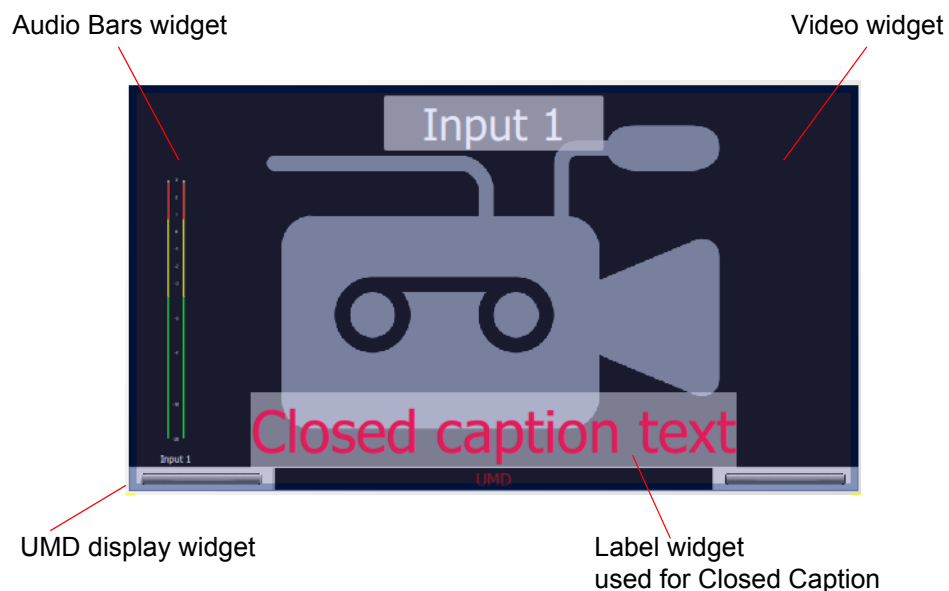


Figure 138 Video Auto-Tile

There are a number of **Behaviours** and **Bindings** that look at various system state or alarm values, apply some logic/mappings and then set widget on-screen properties. For example, to set a border flashing when video input is lost.

Figure 139, Figure 140, Figure 141 and Figure 142 show the **Behaviour** and **Bindings** framework for each of the video **Auto-Tile** widgets.

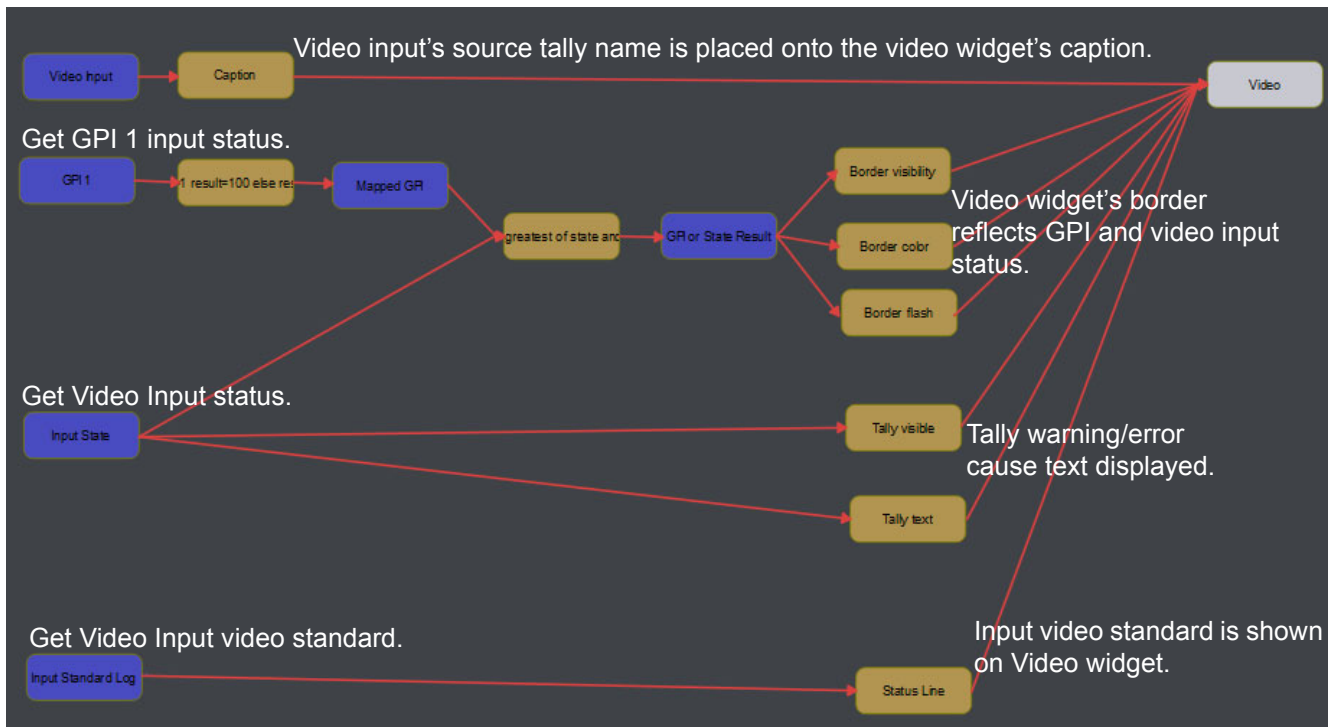


Figure 139 Video Auto-Tile - Video Widget's Behaviours and Bindings

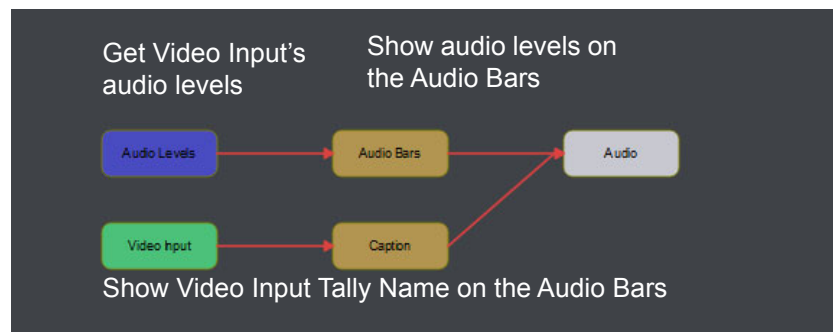


Figure 140 Video Auto-Tile - Audio Bars Widget's Behaviours and Bindings

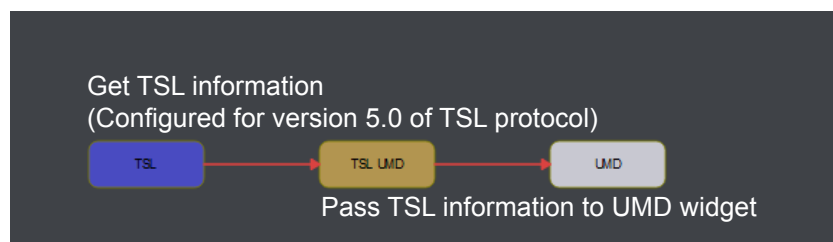


Figure 141 Video Auto-Tile - UMD Widget's Behaviours and Bindings

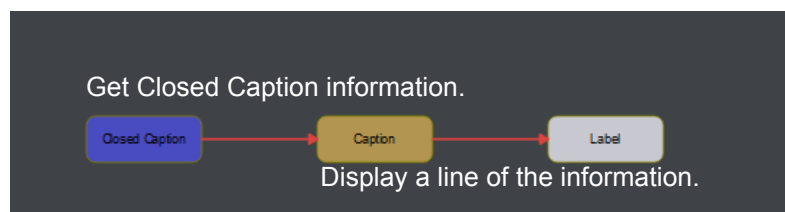


Figure 142 Video Auto-Tile - Label Widget's (Closed Caption) Behaviours and Bindings

6 Groups of Widgets

Groups of Widgets

| | |
|---|-----|
| Widget Groups | 147 |
| Create a Group | 147 |
| Undoing a Group (Ungroup) | 149 |
| Selecting and Moving a Group | 150 |
| Removing a Widget from a Group | 151 |
| Adding a Widget to a Group | 152 |
| Resizing a Group and Group Layout | 153 |
| Properties of a Group | 154 |
| Layout Property Examples | 156 |
| Adding / Moving a Widget and Automatic Repositioning of Group Items | 158 |
| Group Layout | 160 |
| Spacers and Group Layouts | 160 |
| Widget Layout Item Properties | 163 |
| Group Layout Example - Soft Panel | 165 |

The descriptions below use examples requiring the selection/deselection of various on-screen items in the **Tile Editor**. The following note is a reminder about the various select/deselect tools in Orbit.

Note:

Selection/ Deselection of graphical items when editing items in the Tile Editor:

| Select / Deselect | Action |
|------------------------------|---|
| Deselect All | Click outside the schematic area. Or press the ' Esc ' keyboard key. |
| Select | Click on an item to select it. Alt-Click selects another item: <ul style="list-style-type: none"> • In a stack of items. • Or selects the Group Layout that an item belongs to. |
| Add Select | Shift-Click on an item to add it to current selection. Shift-Alt-Click to add another item from a stack of items to the current selection. |
| Remove from Selection | Shift-Click on an <i>already-selected</i> item to remove it from the current selection. |
| Select All | 'Ctrl-A' selects all items. |
| Select an Area | Shift drag cursor on screen to describe a rectangular area. All objects wholly or partly in the area will be selected. |

6.1 Widget Groups

In Orbit, widget items can be grouped together on a **Tile** or on a **Schematic**. Grouping allows a user to control the appearance of the layout of grouped items on-screen (**Group Layout**) when the group is re-sized.

The Group/Ungroup control is in the right-click menu of the **Tile Editor**; the menu item appears when two or more widgets are selected.

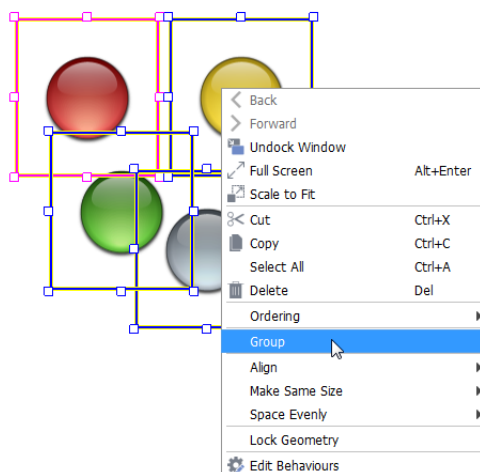
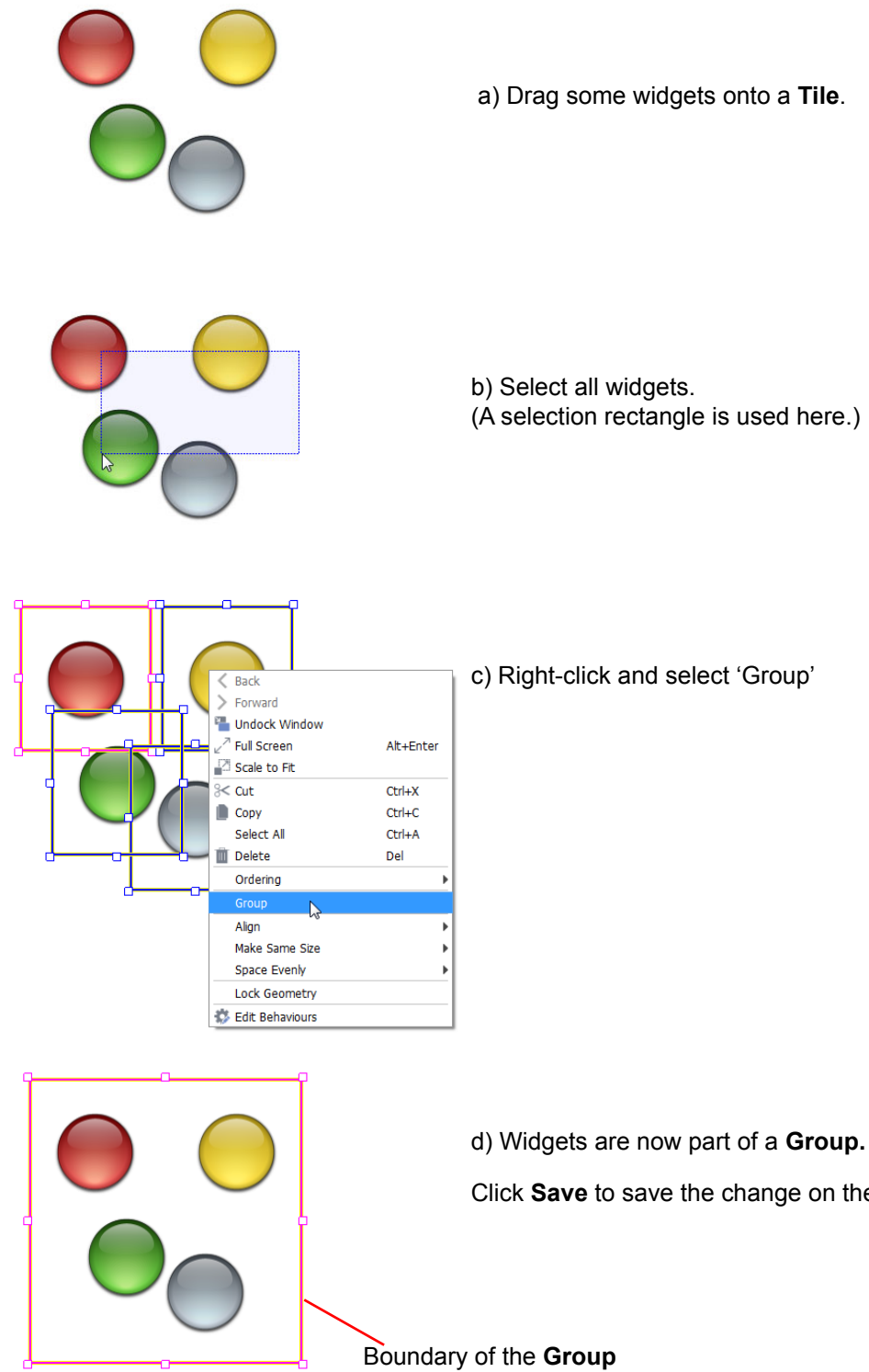


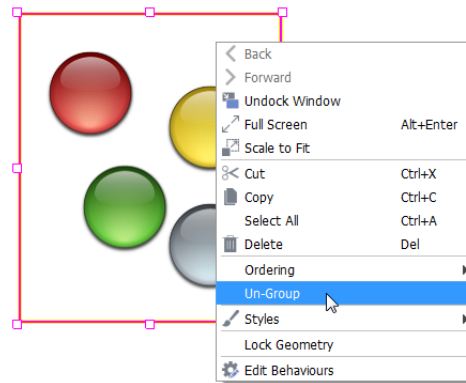
Figure 143 Group / Ungroup Control

6.1.1 Create a Group

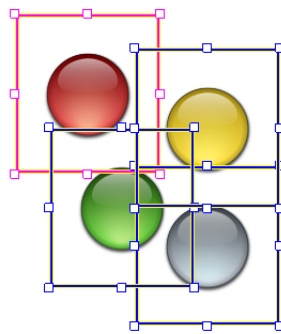
To create a **Group**:

**Figure 144 Creating a Group**

6.1.2 Undoing a Group (Ungroup)



a) Select the **Group**.
Right-click and select 'Ungroup'.



b) The **Group** is removed.

Click **Save** to save the change on the **Tile**.

The individual widgets are not now grouped together.

Figure 145 Undoing a Group (Ungroup)

6.1.3 Selecting and Moving a Group

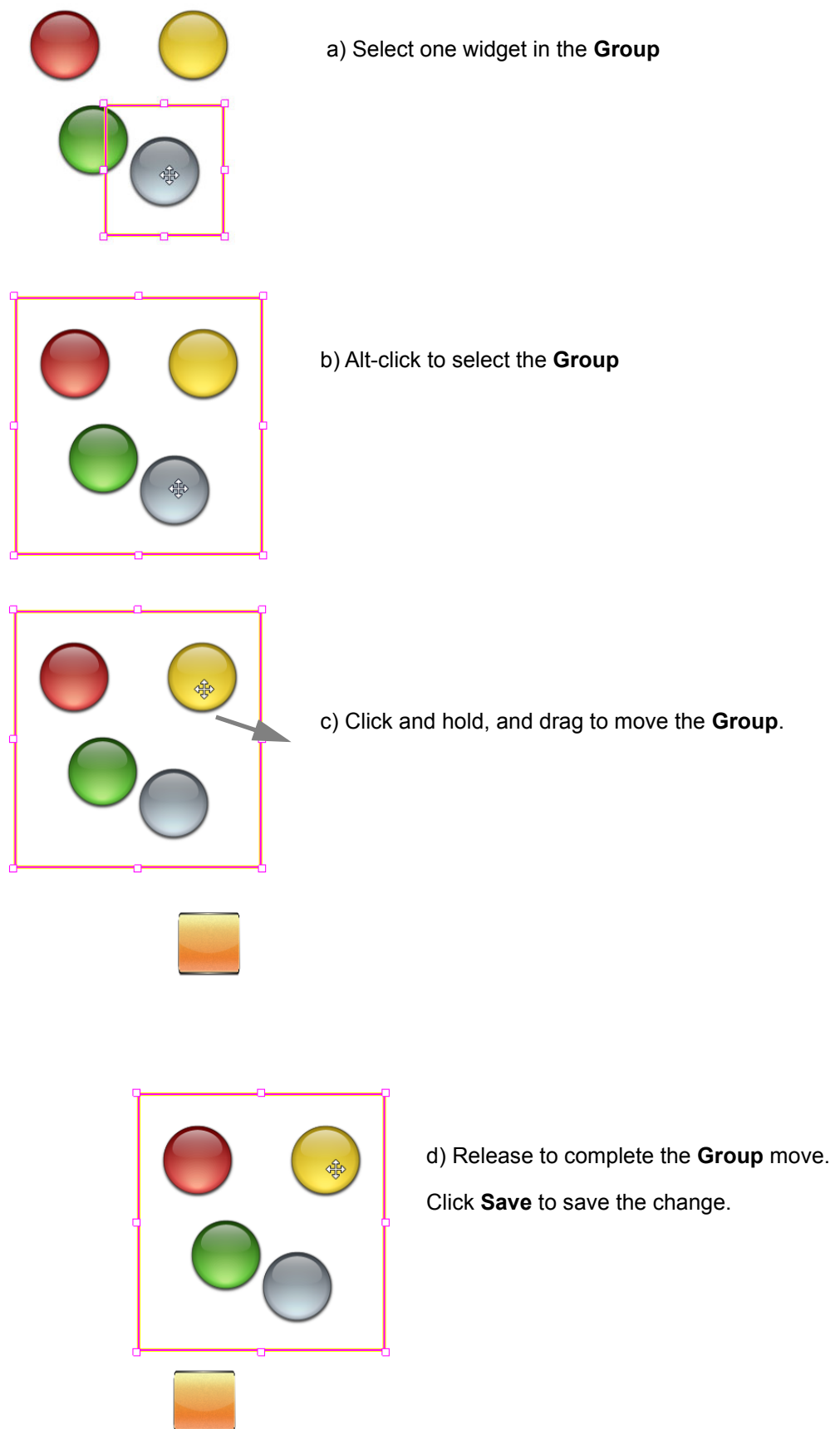
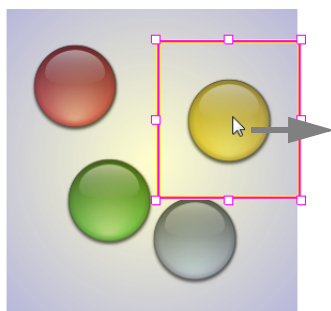


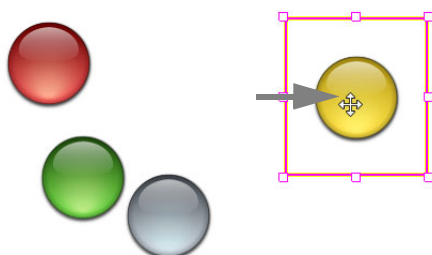
Figure 146 Moving a Group

6.1.4 Removing a Widget from a Group

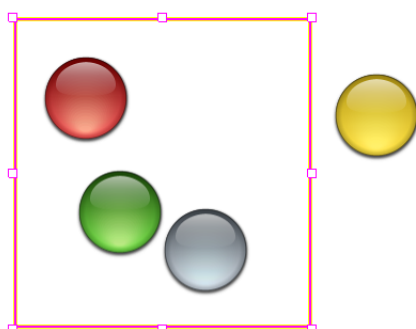


a) Select one widget in the **Group**.
Click and begin to drag the widget.

The extent of the **Group** is shown in gray on the screen.



b) Move the widget outside of the **Group**.



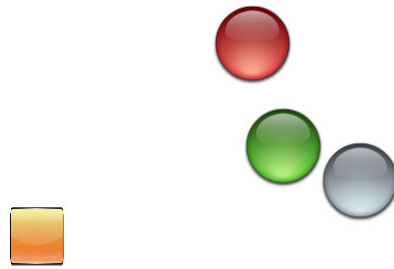
c) Release to complete the widget move.

The widget is now outside the **Group**
and no longer part of it.
The widget has been removed from the
Group.

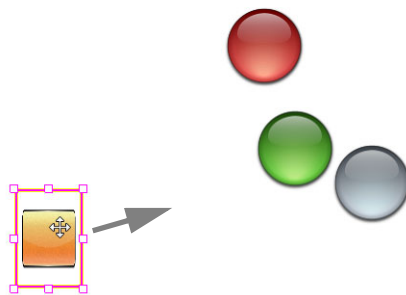
Click **Save** to save the change.

Figure 147 Removing a Widget from a Group

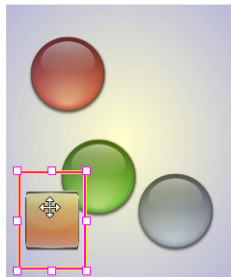
6.1.5 Adding a Widget to a Group



a) If the widget to be added is not on the **Tile** or **Schematic**, drag the widget into the **Tile Editor**.

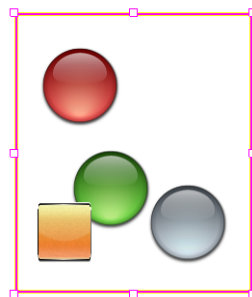


b) Select the widget to be added to a **Group**.



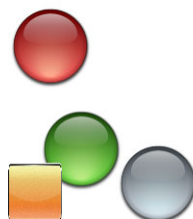
c) Drag the widget into the **Group**.

The gray extent of the **Group** is shown as the dragged widget approaches the **Group** on the **Tile Editor** screen.



d) Release the widget to complete the widget move.

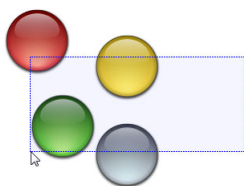
The widget is now inside the **Group** and is now part of it.



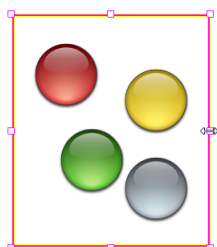
e) The widget has been added into the **Group**. Click **Save** to save the change.

Figure 148 Adding a Widget to a Group

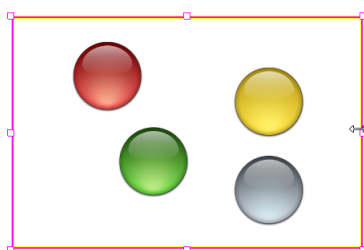
6.1.6 Resizing a Group and Group Layout



a) Select the **Group**.
(A selection rectangle is used here.)

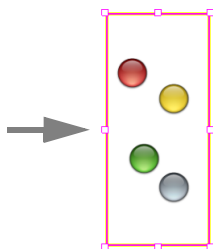


b) The **Group** is selected.

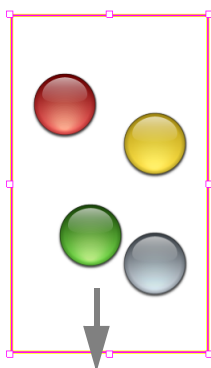


c) Drag a side of the **Group** to increase the width.

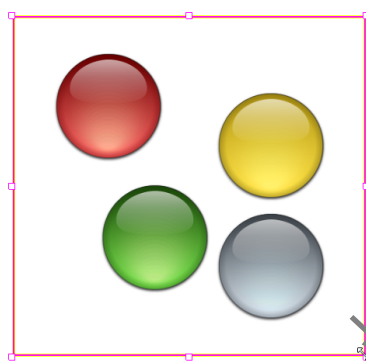
Widget positions within the **Group** are maintained in proportion to the width of the **Group**.



d) Or
drag a side of the **Group** to decrease the **Group** width.



e) Or drag the bottom edge of the **Group** to increase the **Group** height.



f) Or drag a corner of the **Group** to increase/decrease both width and height.

g) Click **Save** to save any change.

Note: Widget positions within the **Group** (**Group Layout**) are maintained in proportion to the size of the **Group**.

Figure 149 Resizing a Group

6.2 Properties of a Group

A **Group** has properties which are visible and editable in the **Tile Editor** when the **Group** is selected. See Figure 150.

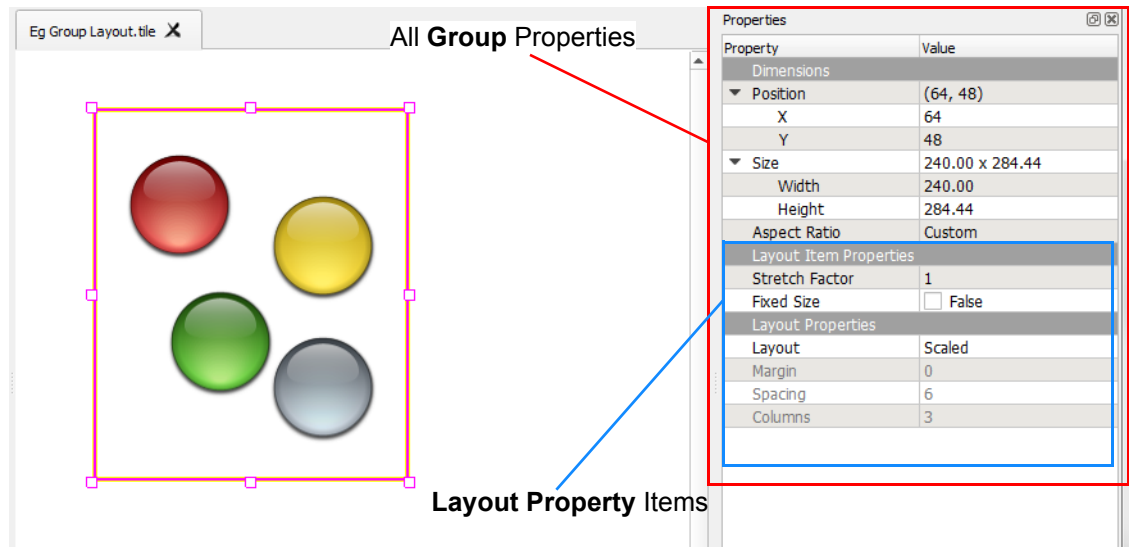


Figure 150 Group Properties

There are properties which control the layout of the **Group** and of the member items of the **Group**. These include properties relating to:

- The layout of the whole **Group** item on-screen when it is a member of *another* group (**Layout Item Properties**):
 - Stretch Factor and Fixed Size.
- The layout of widget items *within* the **Group** itself (**Layout Properties**):
 - Layout, Margin, Spacing and Columns.

Table 23 lists **Layout Item Properties** which take effect when the whole group item is part of *another* group (i.e. a group containing a **Group**).

Table 24 lists **Layout Properties** which control the layout of items *within* the **Group**.

| Item | Description |
|--------------------------------|--|
| Layout <i>Item</i> Properties: | (These are properties only relating to the layout of the whole Group on-screen when the whole Group is itself a member of <i>another Group</i> .) |
| Stretch Factor | <p>Enter scaling factor for the whole Group. This is a scaling factor for the scaling of this whole Group when it is part of <i>another Group</i> (a 'top-level' Group). And the scaling is done in comparison to other items in that <i>other Group</i>.</p> <p>Stretch Factor only applies when the 'top-level' Group's Layout property value is set to: 'Grid', 'Horizontal' or 'Vertical'. It does not apply for 'Scaled'.</p> |
| Fixed Size | <p>Check box.</p> <p>Select to fix the size of the whole Group.</p> <p>Note: Fixed Size overrides Stretch Factor and any scaling. The whole Group remains a fixed size within any 'top-level' Group it is part of.</p> |

Table 23 Layout *Item* Properties of a Group (Layout *Item* Properties)

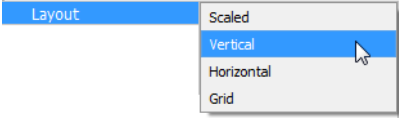
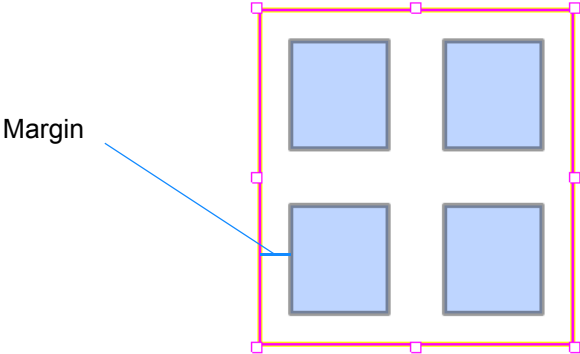
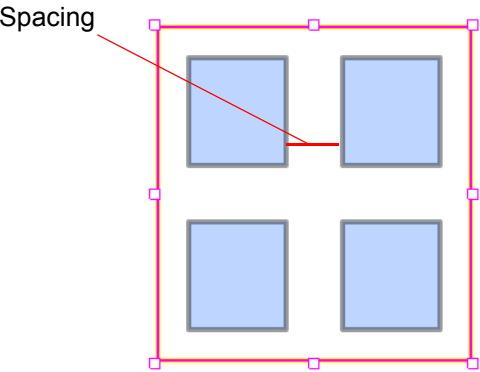
| Item | Description |
|--------------------|---|
| Layout Properties: | <p>(These are properties relating to widget items <i>within</i> the Group.)</p> <p>See Section 6.2.1 “Layout Property Examples” on page 156 for examples of the use of these settings.</p> |
| Layout | <p>Drop-down box.</p> <p>Select the type of layout to apply to the widgets in the Group.</p> <div></div> <p>The layout options are:</p> <ul style="list-style-type: none">• Scaled - A scaled arrangement of how the widgets are positioned in the Group.• Vertical - The widgets are arranged vertically in a column.• Horizontal - The widgets are arranged horizontally in a row.• Grid - The widgets are arranged in a grid formation. <p>The Scaled layout option does not change the relative positions of widgets.</p> <p>The Vertical, Horizontal and Grid layout options do automatically reposition widgets.</p> <p>See Section 6.2.2 “Adding / Moving a Widget and Automatic Repositioning of Group Items” on page 158.</p> |
| Margin | <p>Specify a margin around items in a Group. (Unit = pixels)</p> <div></div> <p>See Figure 152 for examples.</p> <p>(Applicable for Layout Properties > Layout = ‘Grid’, ‘Horizontal’ and ‘Vertical’.)</p> |

Table 24 The Layout Properties of a Group (Layout Properties)

Spacing

Specify the spacing between items in a **Group**. (Unit = pixels)



See Figure 152 for examples.

(Applicable for **Layout Properties > Layout** = 'Grid', 'Horizontal' and 'Vertical'.)

Columns

Specify the number of columns of items in a 'Grid' layout in a **Group**.
See Figure 152.

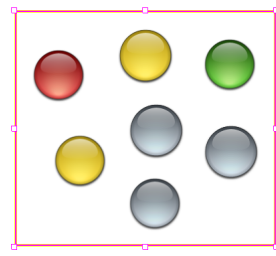
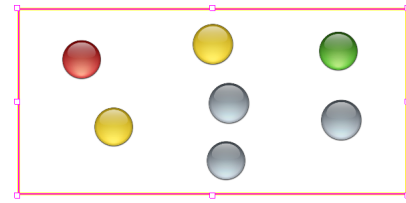
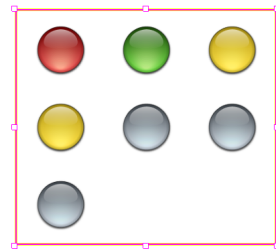
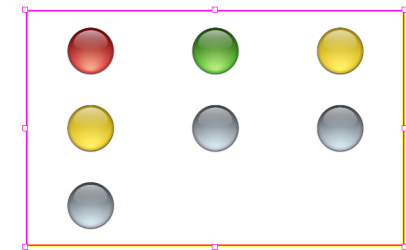
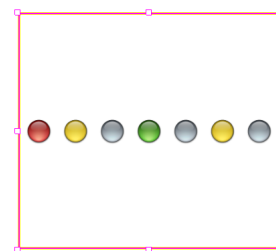
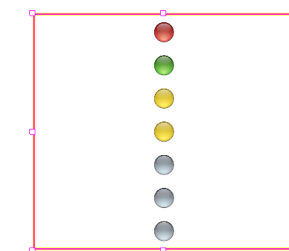
(Applicable for **Layout Properties > Layout** = 'Grid'.)

Table 24 The Layout Properties of a Group (Layout Properties) (Continued)

6.2.1 Layout Property Examples

- Layout Properties > Layout
- Layout Properties > Margin
- Layout Properties > Spacing
- Layout Properties > Columns

The widgets within a **Group** can be assigned a layout (arrangement) with the **Group's Layout Properties > Layout** setting. Figure 151 shows the **Group Layout** applied in each case and Figure 152 shows the effect of some of the other **Layout Property** settings.

a) **Layout** = 'Scaled'.b) **Layout** = 'Scaled' and **Group** is resized.c) **Layout** = 'Grid' and Columns = 3.d) **Layout** = 'Grid' and Columns = 3,
Group is resized.e) **Layout** = 'Horizontal'.f) **Layout** = 'Vertical'.**Figure 151 Group Layouts and Different Settings of Layout Properties > Layout**

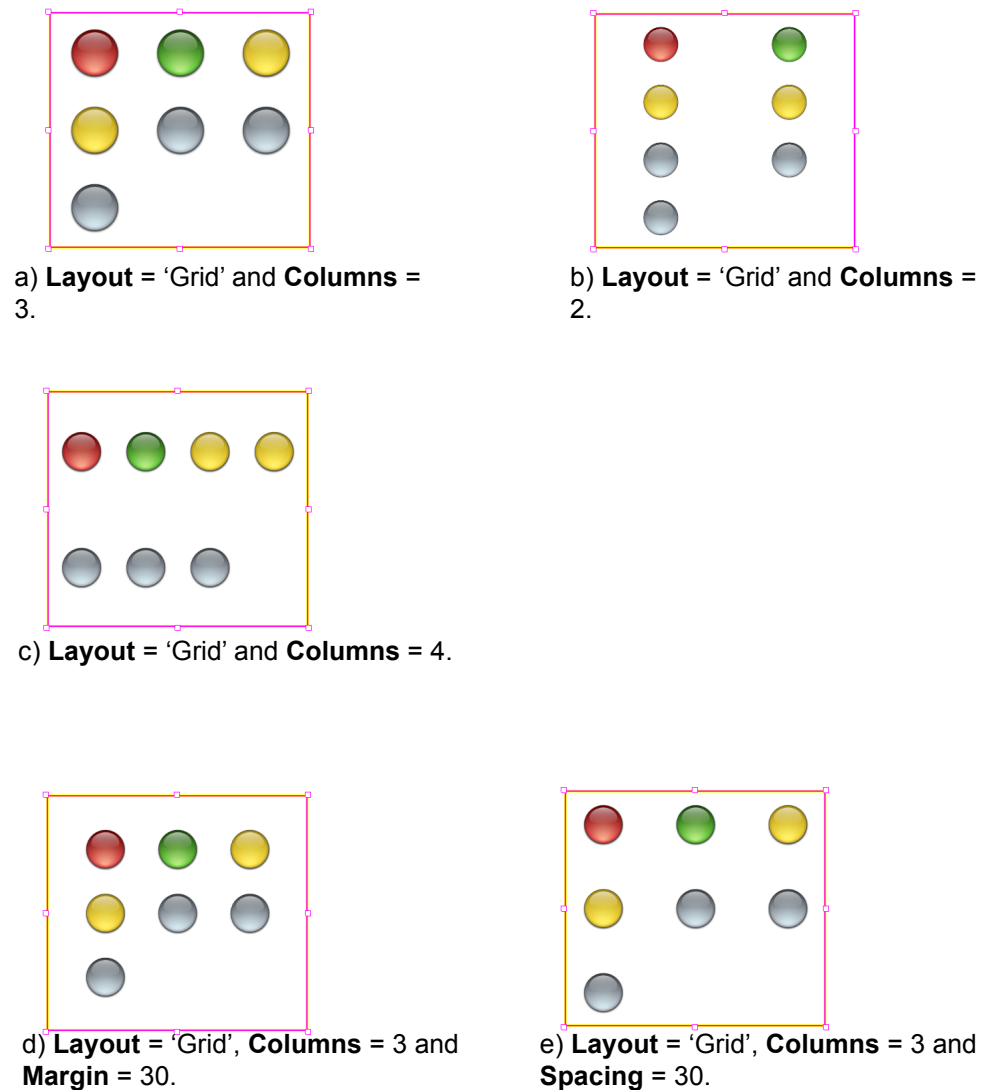


Figure 152 Layout property > Margin, Spacing and Columns Settings

6.2.2 Adding / Moving a Widget and Automatic Repositioning of Group Items

If a widget is added into a **Group** which has its

Layout Properties > Layout setting set to 'Vertical', 'Horizontal' or 'Grid',

then the **Group** members' positions are actively managed on-screen. The group members are automatically repositioned, depending on where the moved/added widget is placed in the **Group**. See Figure 153.

Note: When a widget is added into a **Group** which has its **Layout Properties > Layout** setting set to 'Scaled', *no* automatic widget repositioning is done.

Note:

Items within a **Group** are automatically repositioned when the **Group**'s property:

Layout Properties > Layout = 'Vertical', 'Horizontal' or 'Grid'.

Items within a **Group** are NOT automatically repositioned when:

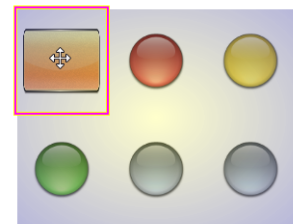
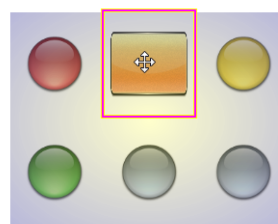
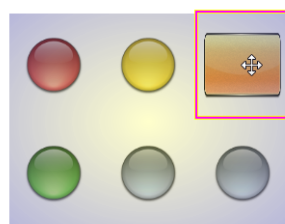
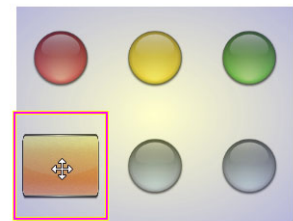
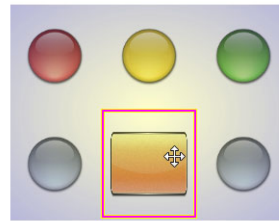
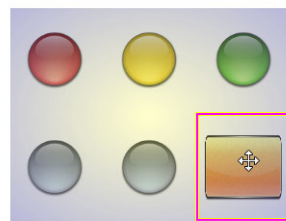
Layout Properties > Layout = 'Scaled'.



a) **Group**

Layout Properties > Layout setting set to 'Vertical', 'Horizontal' or 'Grid'

b) Add a widget



c) Moving a widget in a **Group**, other widgets are automatically repositioned.

Figure 153 Automatic Widget Re-Positioning in a Group

6.3 Group Layout

Widgets and **Groups** can be items in a **Group**.

A **Group** actively controls the on-screen layout of items in the **Group** when the **Group** is being re-sized on a **Tile** or **Schematic**. This provides an active way of controlling the appearance of the **Tile** or **Schematic**. The layout types used are: 'Grid', 'Horizontal' and 'Vertical'.

The following are used to control the layout of items in a **Group**:

- Special widgets, **Spacers**.
See Section 6.3.1 "Spacers and Group Layouts" on page 160.
- Some widget properties.
See Section 6.3.2 "Widget Layout Item Properties" on page 163.
- **Group** property, **Layout Properties > Layout**: 'Grid', 'Horizontal' and 'Vertical'.

(For information about Orbit **Groups**, see Section 6.1 "Widget Groups" on page 147 and Section 6.2 "Properties of a Group" on page 154.)

Figure 154 shows some simple 'Grid', 'Horizontal' and 'Vertical' **Group Layouts**.

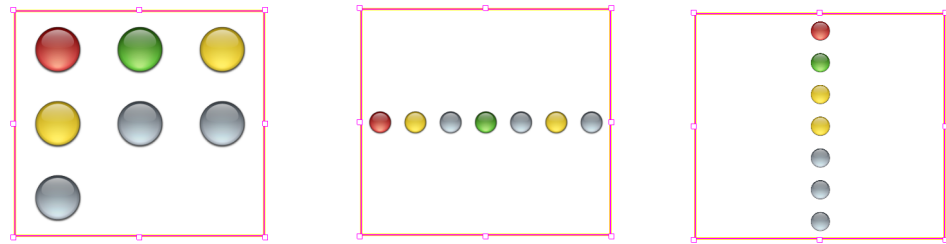


Figure 154 Example Simple Group Layouts

6.3.1 Spacers and Group Layouts

A particular on-screen layout of widgets may be facilitated by using some 'whitespace' widgets, the **Horizontal Spacer** and **Vertical Spacer** widgets. See Figure 155.
(All widgets are listed in Table 37 on page 322 of B).



Figure 155 Spacer Widgets - Horizontal and Vertical

A **Horizontal Spacer** enables relative space to be added between widgets in a horizontal **Group Layout**.

A **Vertical Spacer** enables relative space to be added between widgets in a vertical **Group Layout**.

A typical application for Spacer widgets is when a soft control panel is being designed.

The **Spacer** widgets add space on-screen between other widgets and are invisible on-screen. Figure 156 shows an example use of a **Spacer** widget in a **Group Layout**.

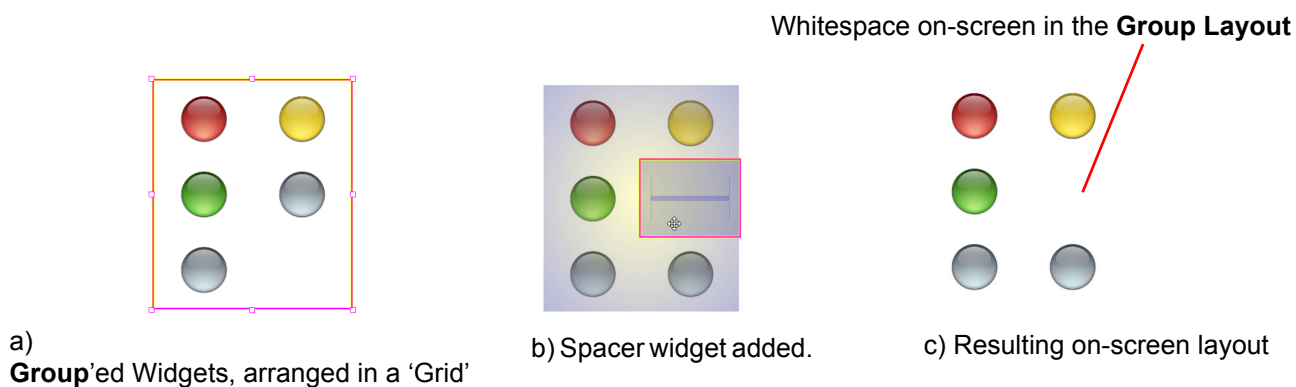


Figure 156 Spacer Widget used in a Group Layout

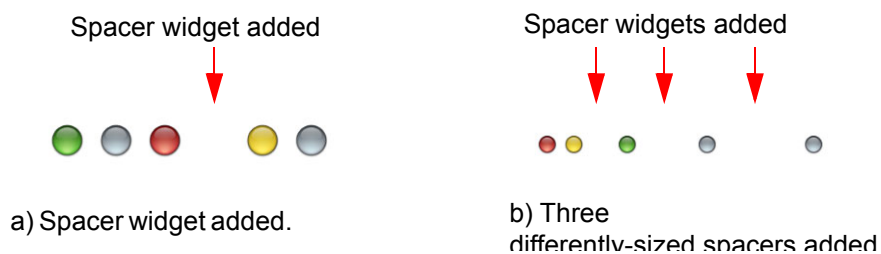


Figure 157 Further Spacer Widgets used in a Group Layout

6.3.1.1 Spacer Widget Properties

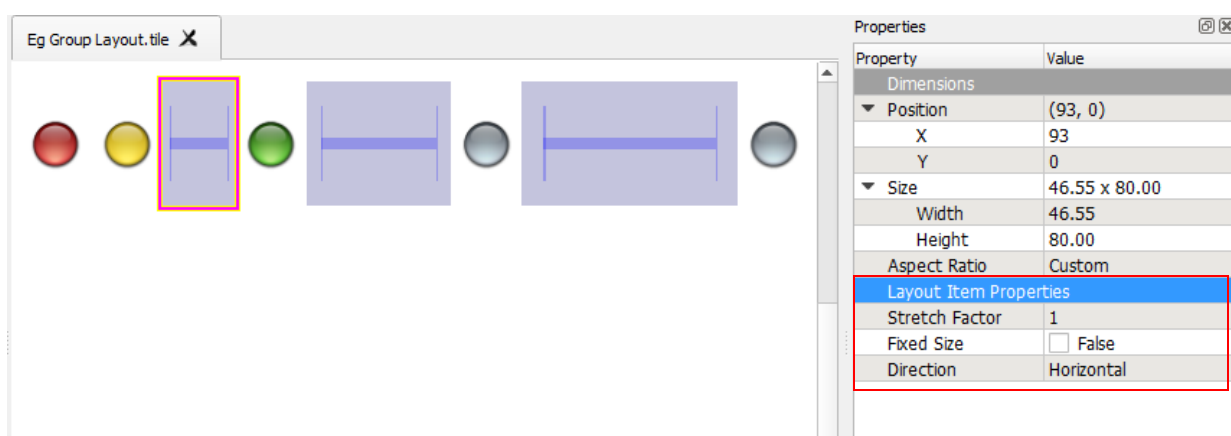


Figure 158 Spacer - Layout Item Properties

| Item | Description |
|-------------------------|---|
| Layout Item Properties: | |
| Stretch Factor | <p>Enter scaling factor for spacer.</p> <p>Scaling is done relative to other items in the Group. See Figure 159a for an example.</p> <p>Stretch Factor only applies for the Group's Layout property values of 'Grid', 'Horizontal' or 'Vertical' layouts, not to 'Scaled'.</p> |
| Fixed Size | <p>Check box.</p> <p>Select to fix the size of the spacer, regardless of any on-screen scaling.</p> <p>Note: Fixed Size overrides Stretch Factor and any scaling. See Figure 159b for an example.</p> |
| Direction | <p>Drop-down box.</p> <p>Select Spacer widget type:</p> <ul style="list-style-type: none">Horizontal.Vertical. |

Table 25 Spacer - Layout Item Properties

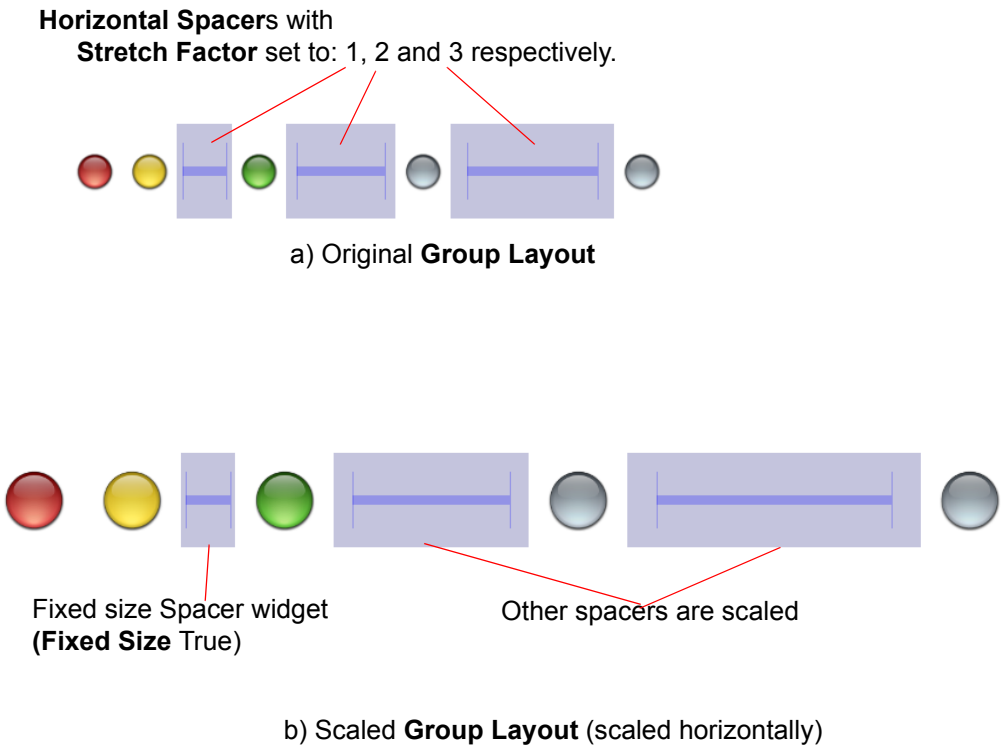


Figure 159 Stretch Factor and Fixed Size (Spacer Layout Item Properties)

6.3.2 Widget Layout Item Properties

A particular on-screen layout of widgets may be realized with the **Layout Item Properties** available on widgets in a **Group**. These properties are similar to those on the Spacer widget properties.

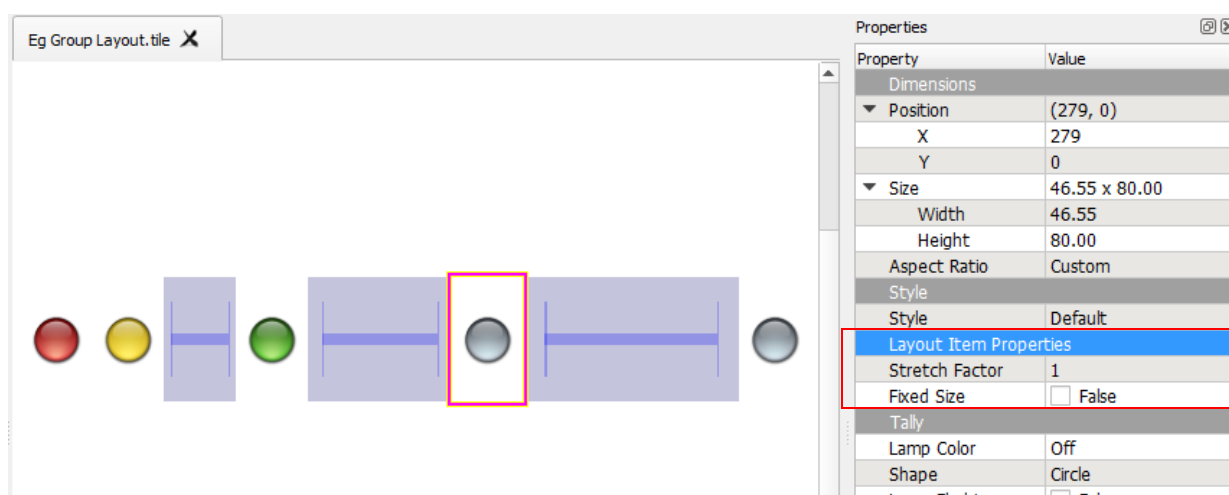


Figure 160 Widget - Layout Item Properties

| Item | Description |
|-------------------------|---|
| Layout Item Properties: | |
| Stretch Factor | Enter scaling factor for the widget. Resizes the widget relative to its original size and other widgets in the Group . Stretch Factor only applies to Group Layout property values of 'Grid', 'Horizontal' or 'Vertical' layouts, not to 'Scaled'. See Figure 161. |
| Fixed Size | Check box. Select to fix the size of the widget, regardless of any on-screen scaling. Note: Fixed Size overrides Stretch Factor and any scaling. The widget item remains a fixed size within any group it is part of. See Figure 162. |

Table 26 Widget - Layout Item Properties

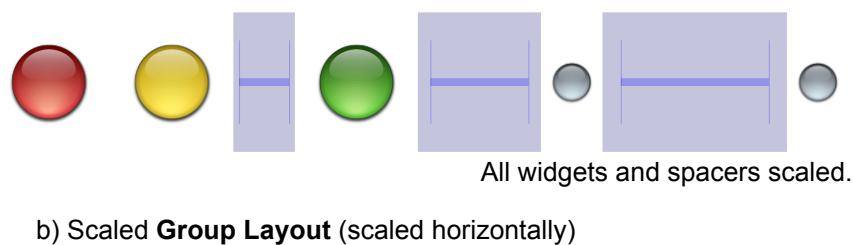
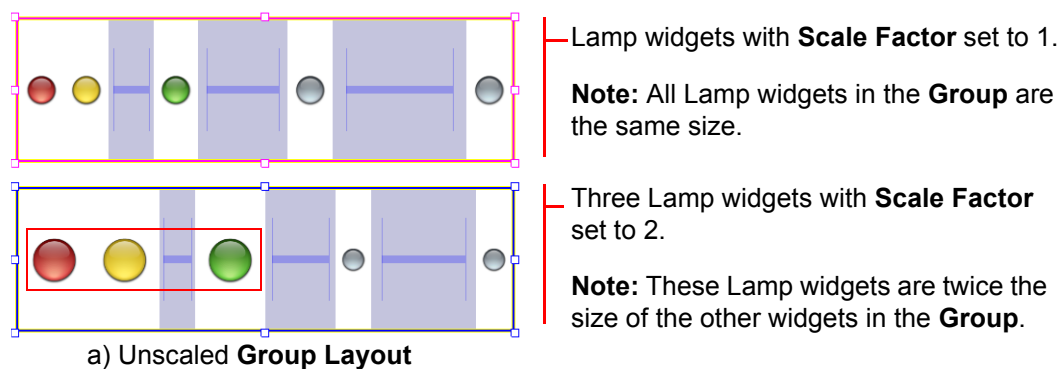


Figure 161 Scale Factor (Widget Layout Item Property)

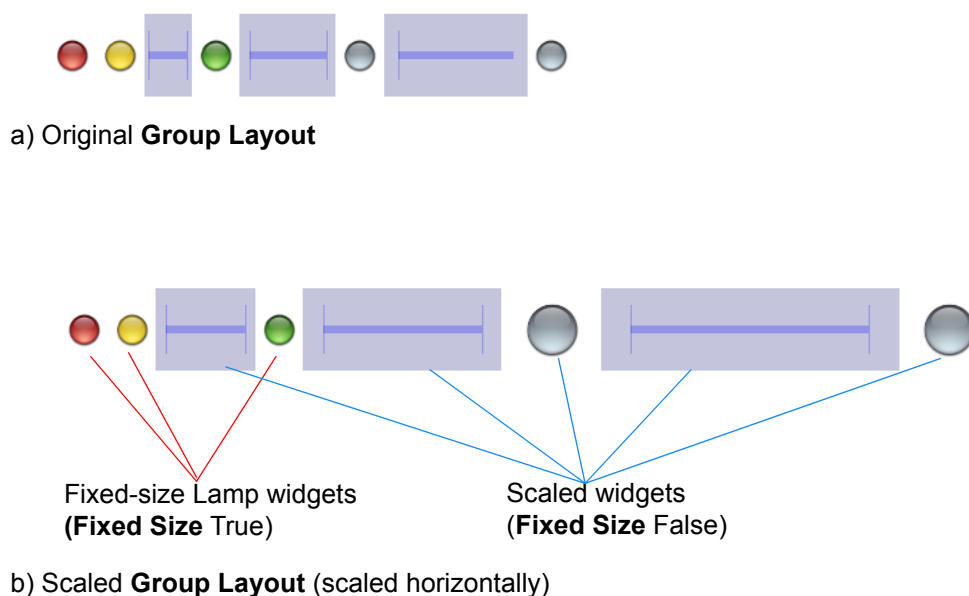


Figure 162 Fixed Size (Widgets Layout Item Property)

6.4 Group Layout Example - Soft Panel

A small soft panel may be designed for controlling:

- Timers on a multiviewer.
- Multiviewer **Wall** layouts.
- Part of an Orbit MapView control/status panel.

The example in Figure 163 shows a basic soft control panel example that has been laid out in Orbit.

The whole soft panel is a **Group** which comprises the following on-screen member items:

- A **Group** comprising two Tally Lamp widgets and a Horizontal Spacer widget.
- A Vertical Spacer widget.
- A **Group** comprising three Button widgets.

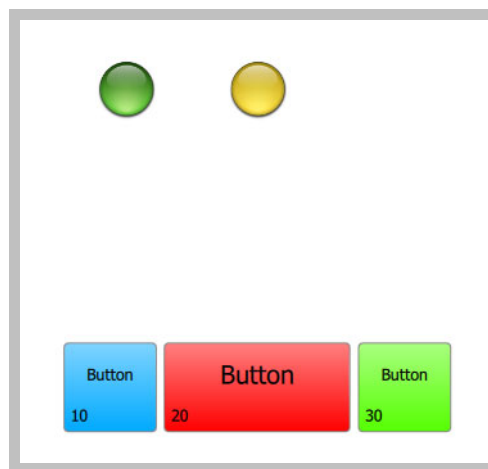


Figure 163 Example Soft Control Panel

Figure 164 shows the member items of the top-level **Group** of the soft panel example.

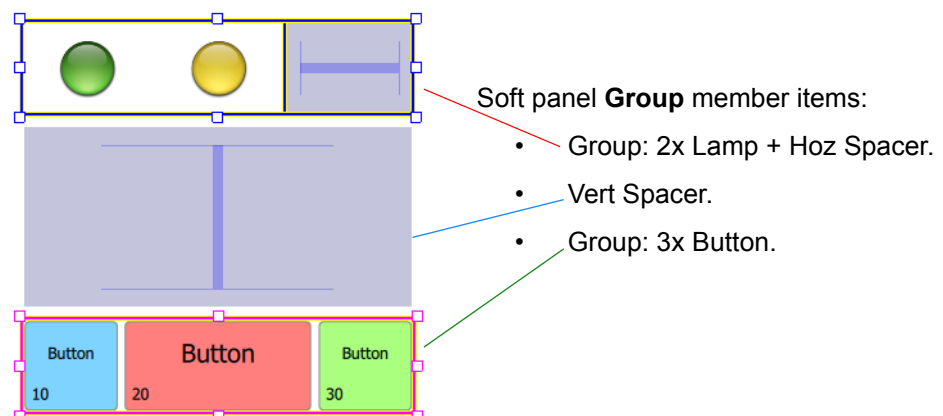


Figure 164 Soft Control Panel's Group Member Items

Figure 165 shows the member items' **Stretch Factor** setting.

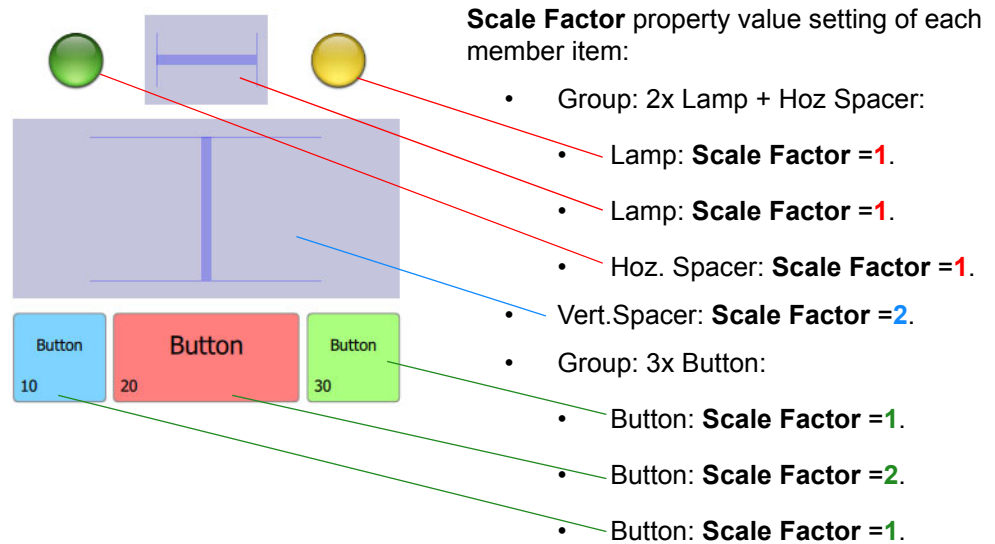


Figure 165 Example Soft Control Panel's Group Layout Features

7 Project Variables

Project Variables

| | |
|---|-----|
| Introduction to Project Variables | 168 |
| Application Examples for Variables | 168 |
| How to Create a User Project Variable | 169 |
| Using a User Project Variable | 170 |
| Tile Variables | 175 |
| Right-click 'Variables...' Menu Item | 175 |
| Set Variables Icon and User Tile Variables | 178 |
| User Tile Variable and Set Variables Example: | 181 |
| 'Create Variable from Property' Example | 184 |

7.1 Introduction to Project Variables

Variables can be used in a Orbit project to include numbers or values in widget or **Behaviour** properties.

The types of variables in a Orbit project are:

- Internal project variables.
- User project variables.
- User tile variables.

Variable names are entered in property value fields when editing video tiles or widgets. In this case, the variable's name is placed in curly-braces.

For example: {My_Address}

Orbit has some **Internal Project Variables** which it maintains itself.
For example:

- **Input** - Multiviewer video input number. These are assigned to video widgets on a video wall. The variable is evaluated at run time.
- **Address** - RollCall address of the target Multiviewer unit. It is evaluated at run time.

These variables may appear in various **Variables** dialogs in Orbit.

A user can define a **User Project Variable** to be used in a project.
See Section 7.1.2 "How to Create a User Project Variable" on page 169.

A user can define a **User Tile Variable** to be used on a tile in a project. These can be defined at wall level.
See Section 7.2.2 "Set Variables Icon and User Tile Variables" on page 178.

7.1.1 Application Examples for Variables

Application example 1: **User Project Variable.**

Define a user project variable called "*TV_Company*" and set its value to be "XYZ".

Widget properties can then be set to use the user project variable "*TV_Company*" to form on-screen labels such as, "XYZ Sports", "XYZ Movies" and "XYZ News 24" etc.

Application example 2: **User Project Variable.**

Define a user project variable called "*My_RollCall_Address*" in an Orbit project.

This variable may then be used throughout an Orbit project everywhere that the RollCall address of a product unit is needed. For example, in a property value.

The "*My_RollCall_Address*" variable can then be changed at run time.
This change will then ripple through the Orbit project automatically, everywhere that the "*My_RollCall_Address*" project variable is used.

Application example 3: **User Tile Variable.**

Declare a user tile variable called "*My_Clock_Label*" to label a clock widget on a tile.

The tile may be used more than once in a wall design. Each time the tile is used, a new value of "*My_Clock_Label*" can be set for that use of the tile. Thus, for example, the same clock can be used in a wall more than once and configured, at wall level, to show "London", "Paris" and "New York".

7.1.2 How to Create a User Project Variable

User Project Variables are defined at project-level. The scope of the variable is across the project. They may be used on **Walls** or on **Tiles** or **Widgets**.

1. Click on **Project > Edit Variables** in the Orbit screen's main menu. The **Variables** dialog is displayed. See Figure 166.

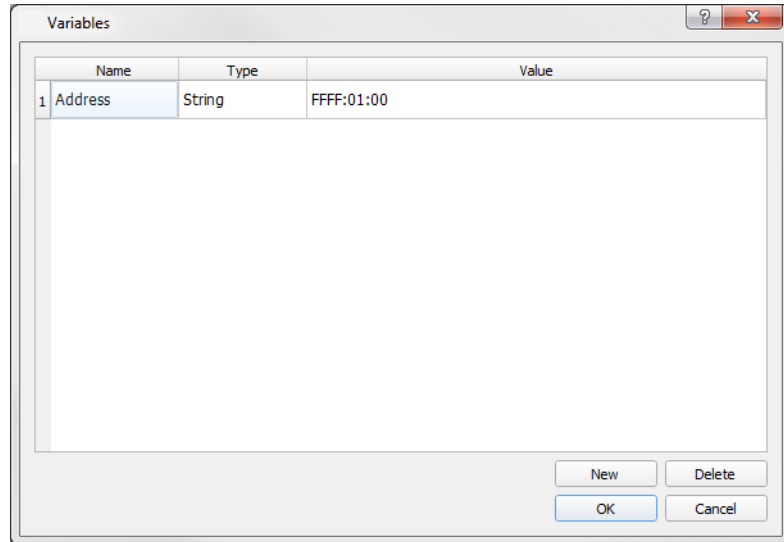


Figure 166 Variables Dialog

2. Click **New**. A new blank entry item appears in the variables list. See Figure 167.

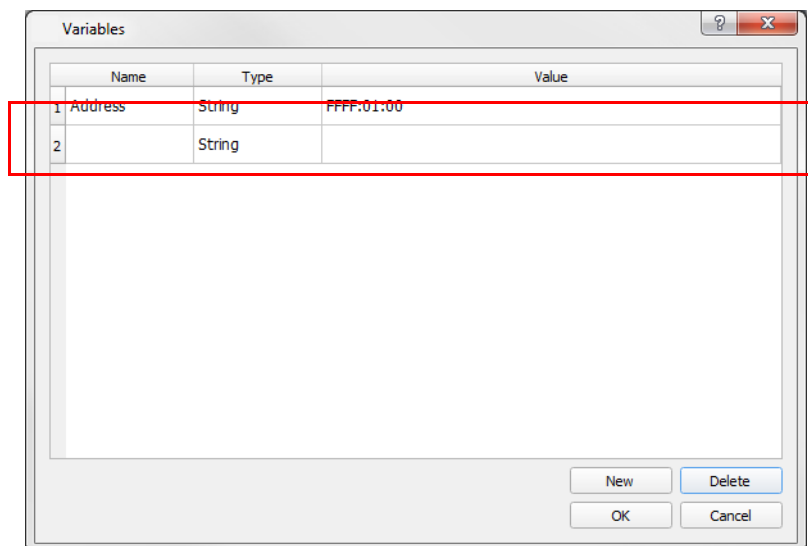


Figure 167 New Blank item in Variables Dialog

3. Enter a **Name** and enter a **Value** for the new variable.
See Figure 168.

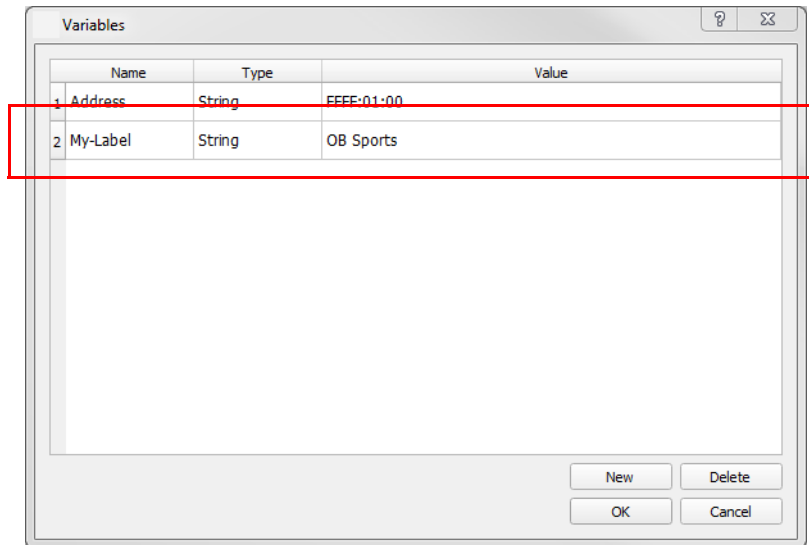


Figure 168 New Blank item in Variables Dialog

A new user project variable has been created in the Orbit project.
The new variable's *value* will be used wherever its **Name** is used in the Orbit project.

For example:

If a **User Variable** named 'My-Label' has the *value* 'OB Sports',
then an on-screen text label with a caption defined as 'My TV {My-Label}'
will show on-screen as:

My TV OB Sports

when the project is pushed to a device running live.

7.1.3 Using a User Project Variable

User project variables may be used in widget properties or in the properties of a Orbit **Behaviour**.

The example here uses a **Label** widget in a **Tile** with a user project variable:

1. Set a project variable **My-Label** to be 'Hello World!'.
2. Add a **Label** widget to a **Tile**.
3. Open the **Tile** in the **Tile Editor**.

4. Select the **Label** widget to display its properties on the right-hand side.
See Figure 169.

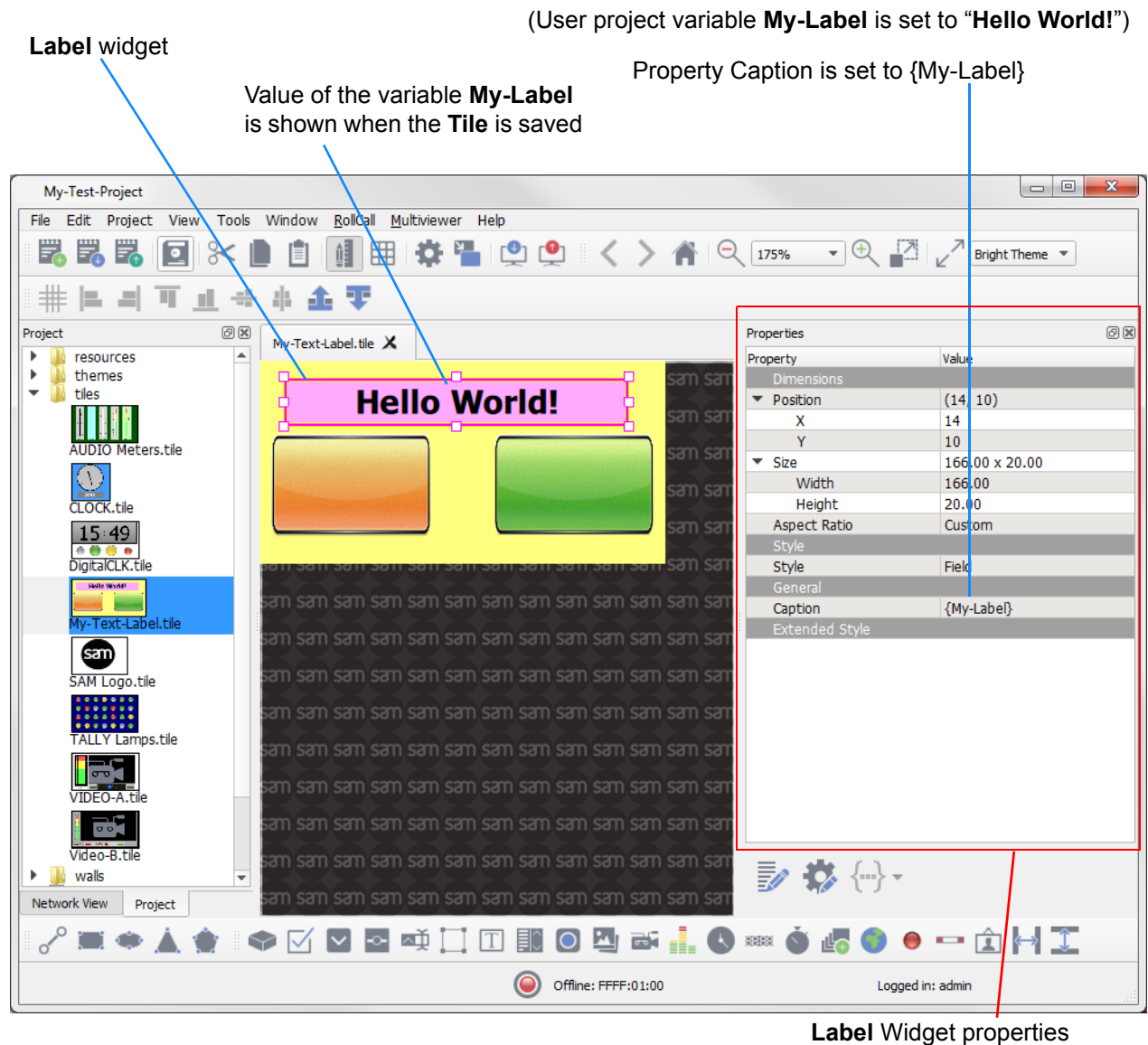


Figure 169 Label Widget on a Tile - Caption Text using {Variable Name}

5. Change the value of the **Caption** property to be a user project variable. I.e. the variable name surrounded by curly braces, see Figure 169.
6. Click the **File Save** icon to save the tile.
The value of the variable is now shown by the text label on the **Tile**.

7. Drag the **Tile** onto a **Wall**.
Save the Wall. (Click the **Save File** icon.)
 The text label now shows the variable's value on the **Wall**. See Figure 170.

Variable's value is shown on the **Wall**

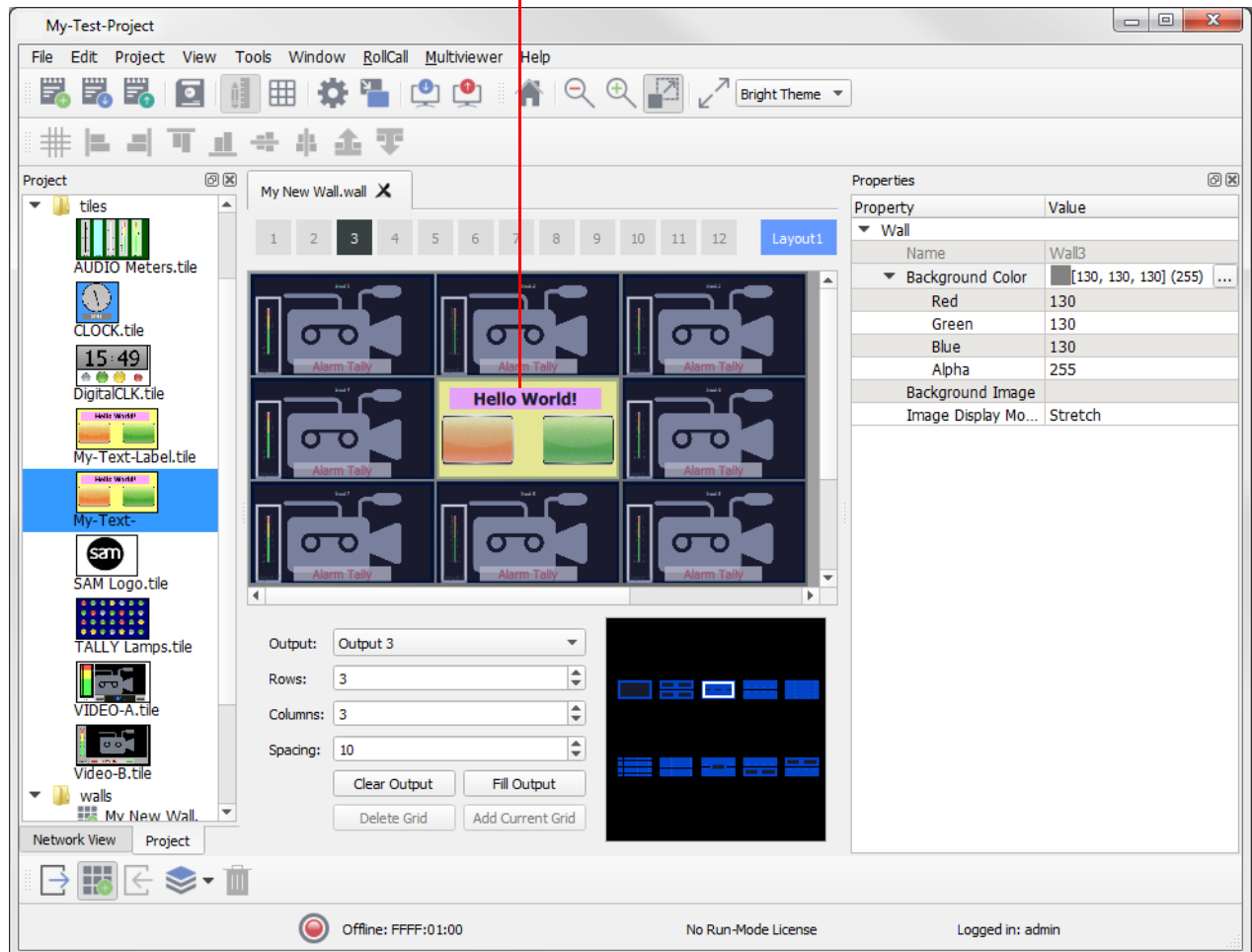


Figure 170 Text Label Widget on a wall

In the example above, a user project variable name was simply used in a property value, i.e. {My-Label}.

Variable names can also be concatenated with each other and with text characters to form compound property values.

For example:

,See Figure 171, define the following user project variables:

- **My-Hello** - defined to be "HELLO".
- **My-Space** - defined to be a space character, " ".
- **My-World** - defined to be "World".

And then the following (compound) value is entered for a Label caption property value:

{My-Hello}{My-Space}{My-World}. Happy?

This shows the following on a wall:

HELLO World. Happy?

1) Define **User Project Variables****My-Hello** - "HELLO"**My-Space** - a space character**My-World** - "World"

| Variables | | |
|------------|--------|--------------|
| Name | Type | Value |
| 1 Address | String | FFFF:01:00 |
| 2 My-Hello | String | HELLO |
| 3 My-Label | String | Hello World! |
| 4 My-Space | String | |
| 5 My-World | String | world |
| 6 My-world | String | GLOBE :-) |

2) Use in **Caption** property value:

General
Caption {My-Hello}{My-Space}{My-World}. Happy?

3) Produce an on-screen caption:

My New Wall.wall (*) X My-Text-Label_Compound.tile X

HELLO world. Happy?

Properties

| Property | Value |
|----------------|--|
| Dimensions | |
| Position | (12, 10) |
| X | 12 |
| Y | 10 |
| Size | 176.00 x 20.00 |
| Width | 176.00 |
| Height | 20.00 |
| Aspect Ratio | Custom |
| Style | |
| Style | Field |
| General | |
| Caption | {My-Hello}{My-Space}{My-World}. Happy? |
| Extended Style | |

Figure 171 Concatenating Variables and Text - "HELLO world. Happy?"

Note:**Case Sensitive Variable Names:**

Be careful. Variable names are case sensitive.

Thus, the following two user project variable names are NOT the same:

My-World

My-world

This results in two different user project variables being defined. As shown in Figure 172.

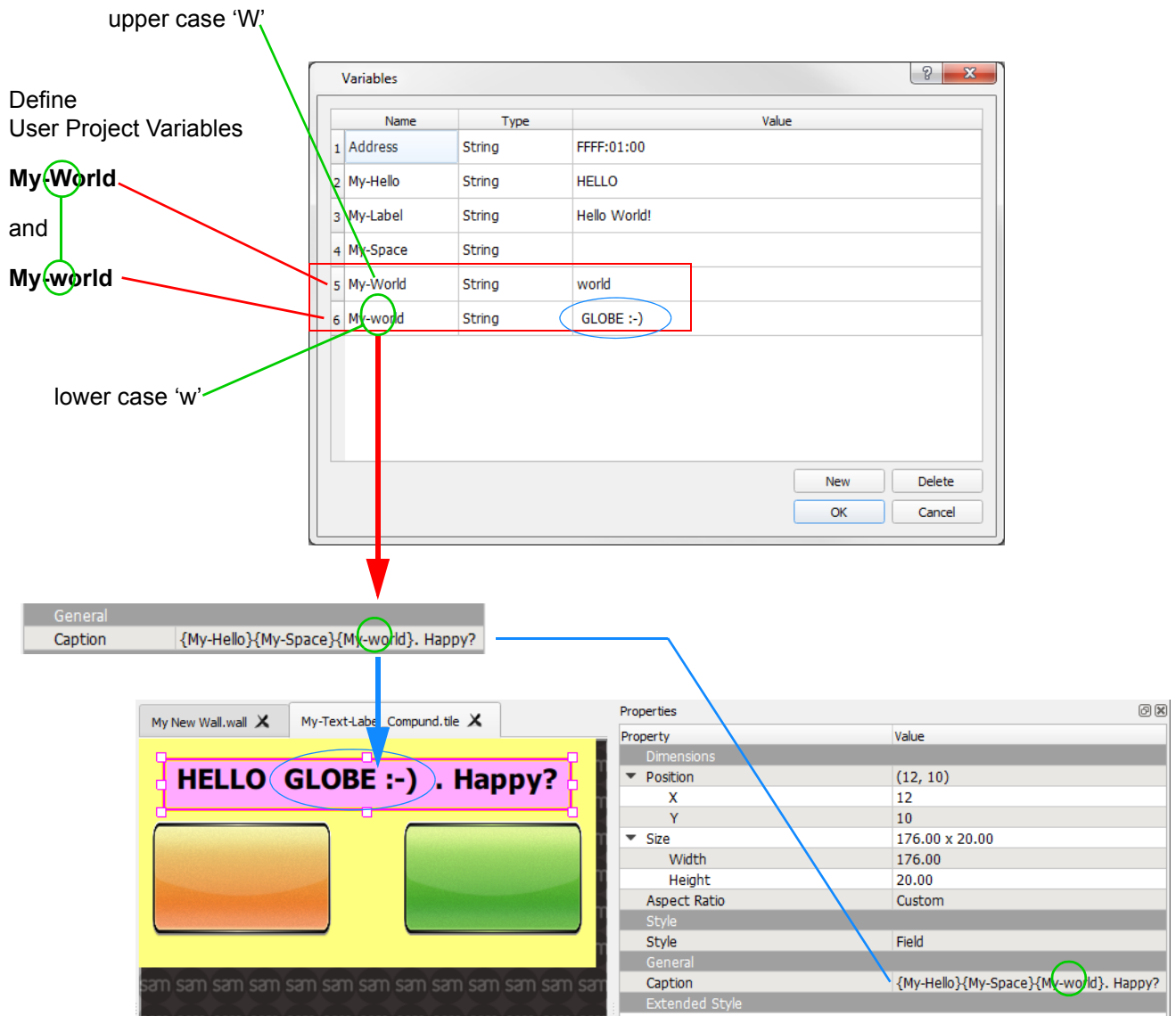


Figure 172 Case Sensitive nature of Variables (variable 'My-world' used)

7.2 Tile Variables

Tile Variables can be created by the user and used on a **Tile**. The scope of these variables is at 'tile-level'. The variables are exposed at 'wall-level' when a **Tile** is deployed on a **Wall** and the variables can be set at wall-level.

7.2.1 Right-click 'Variables...' Menu Item

Tile Variables are accessed through the 'right-click **Variables...**' menu item in the **Tile Editor**. (See Section 3.6 "Schematic 'Right-Click' Context Menus" on page 92.)

A variable can be set up and used at 'tile-level' throughout the **Tile**.

1. In the **Tile Editor** schematic, deselect all widgets and right-click on the **Tile** in the schematic area.
A menu appears. See Figure 173.

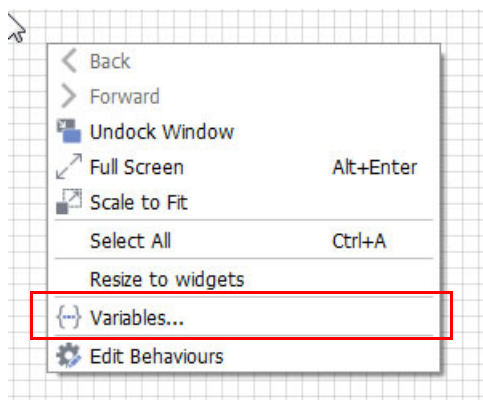


Figure 173 Right-Click Menu

2. Click the **Variables...** menu item.
A Variables dialog appears.
This is the **Tile Variables dialog**. See Figure 174.

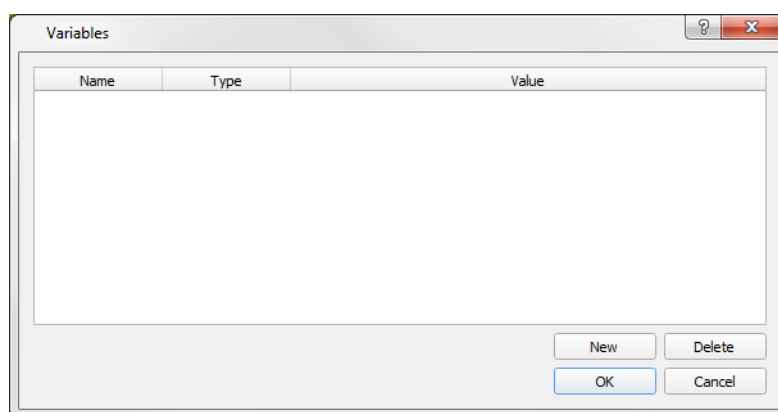


Figure 174 Tile Variables Dialog (empty)

Now define a tile variable to be used on this **Tile**:

3. Click **New**.
A new, empty item appears in the dialog. See Figure 175.

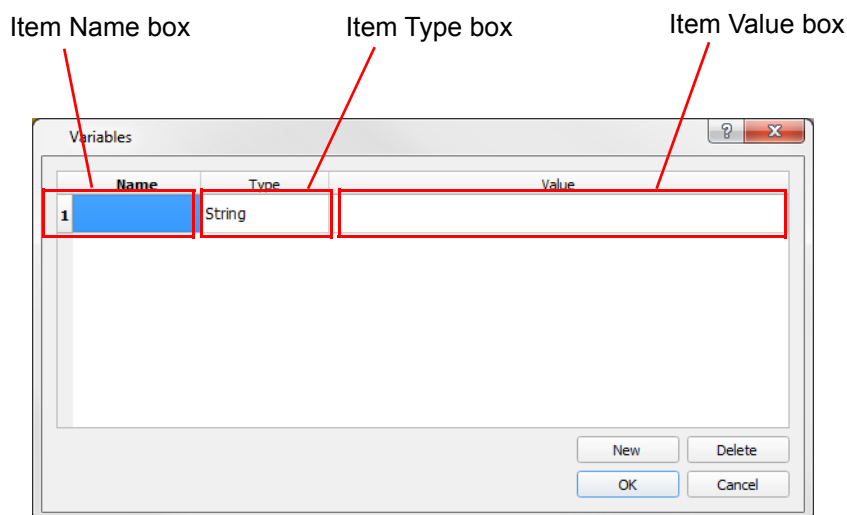


Figure 175 Tile Variables Dialog (new Item)

4. Double-click in the new item's **Name** box and enter a name for a **User Tile Variable**. For example, 'My-Tile-Variable'.
5. Double-click in the item's **Type** box, select a variable type. ("String" for text; "Integer" for a number; "Boolean" for a true/false, checked/unchecked variable type.)
6. Double-click in the item's **Value** box. Enter a value for the **User Tile Variable**.
This value is used as a placeholder until a specific value is assigned at wall level.
7. Figure 176 shows three example **User Tile Variables**.

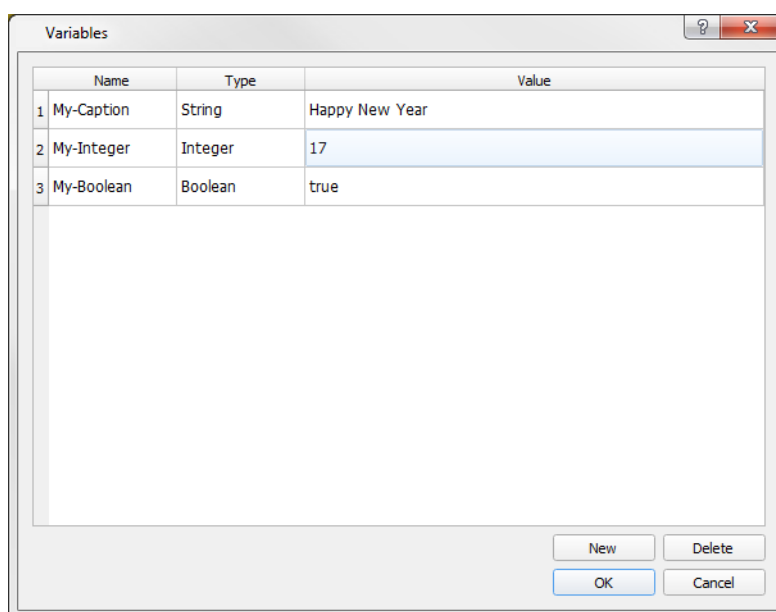


Figure 176 Tile Variables Dialog (three variables defined)

8. Click **OK**. The dialog closes.

Some **User Tile Variables** have been defined for the tile. The variables can be 'string', 'Boolean' or 'integer' data types. They can be assigned a value each time the **Tile** is used on a **Wall**.

'Named Value' Variable:

For the case of an integer data type variable, there are further options in the dialog. When an integer variable type is selected, some more settings are shown, which are used to define a 'named value' data type (enumeration data type). See Figure 177.

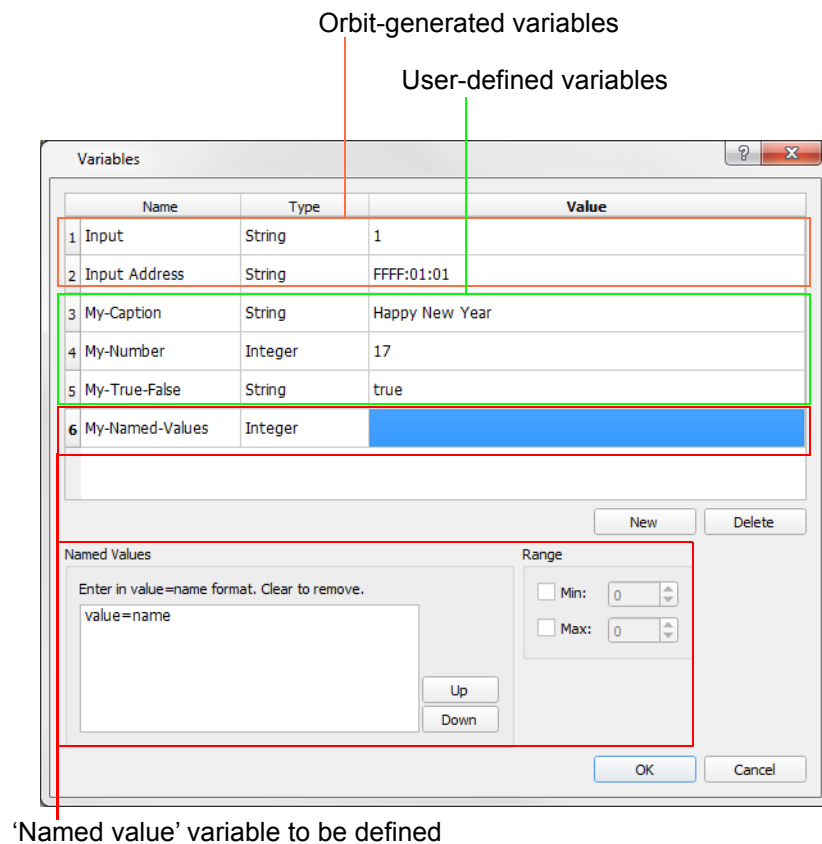


Figure 177 Defining a 'Named Value' User Variable

A 'named value' (enumerated) data type uses an integer data type in a special way:

- A range of integer values are used.
- Values in the range are assigned a name, typically describing what it represents.

In the **Tile Variables** dialog:

- At the **Named Values** text box, double-click in the text box to select the text.
- Enter:
 '0=Red' and press the **Enter** key on your keyboard,
 '1=Green' and press the **Enter** key on your keyboard,
 '2=Amber' and press the **Enter** key on your keyboard.

See Figure 178, in the screenshot:

- The new variable 'My-Named-Values' has the integer range 0 to 2.
- The values 0, 1 and 2 are in the range and are assigned names: 'Red', 'Green' and 'Amber' respectively.

- Click **OK**.

The defined user variable could be used on a **Tile** to determine the color of some 3-color status indicator (built with a widget).

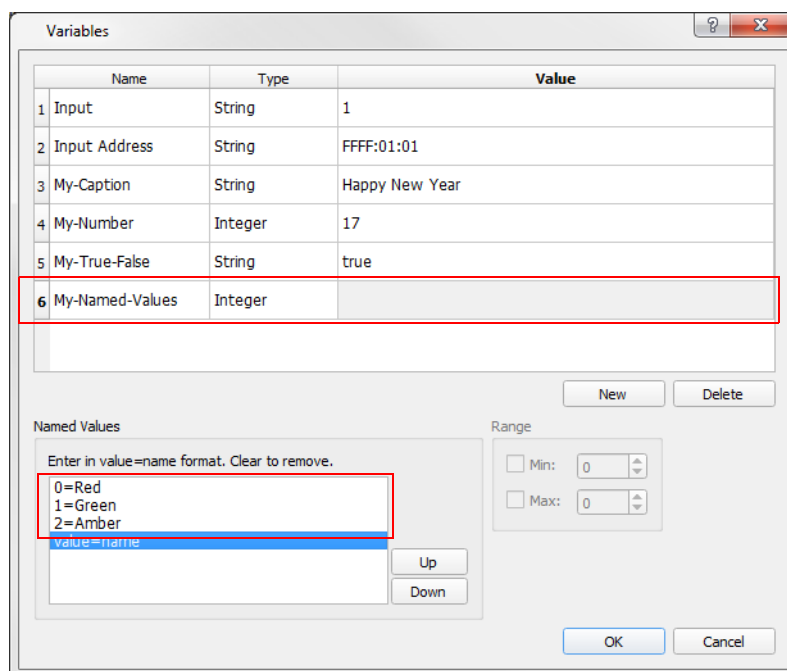


Figure 178 Named Value Variable Defined

7.2.2 Set Variables Icon and User Tile Variables



Figure 179 Advanced Tool Bar - Set Variables icon

The **Set Variables** function allows a widget property to be linked to a **User Tile Variable**. These variables can be assigned a value at the wall level. Thus a tile can be used many times on a wall, with different values set for the **User Tile Variable**.

For example:

A 'UMD tile' can be designed once with a UMD widget or set of widgets, using a variable 'TSL Address'.

The tile can be used many times on a **Wall** and each time a different 'TSL address' can be set for it at 'wall level'.

Figure 179 shows the various sub-menus and dialogs accessed from the **Set Variables** icon.

Tile Editor's Advanced Tool Bar:

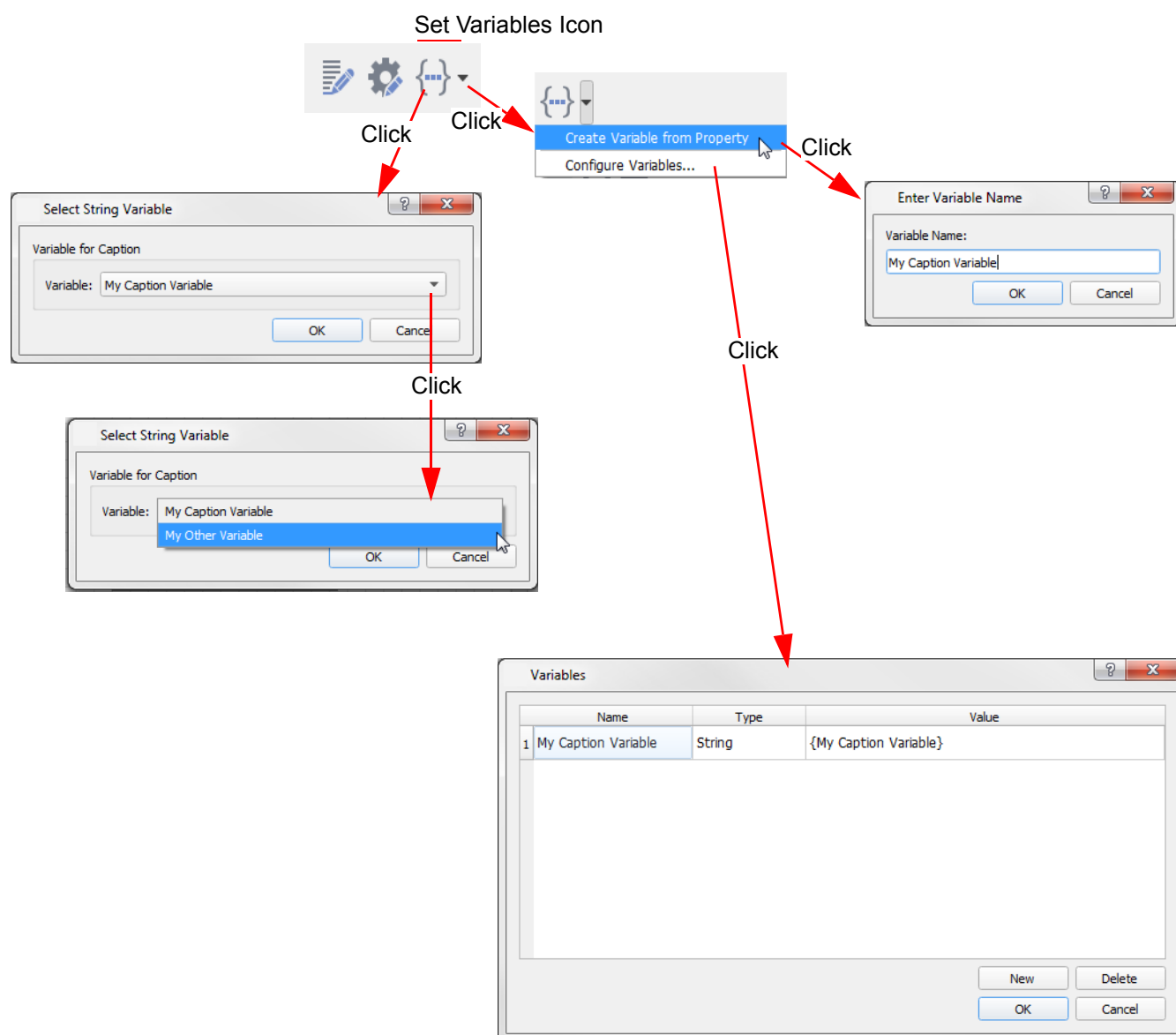




Figure 180 Set Variables Icon's Sub-Menus and Dialogs

Icon availability on-screen:

Some or all of the icon is grayed out depending on whether a variables has been already set. This is explained below:

When a widget is selected in the **Tile Editor** and a widget property value is selected in the **Properties Box**:

- **First Time** - If a variable is being set up for a widget property for the *first time*, then the full **Set Variables** icon is not available. Instead, the down-arrow is only available.
 - Click the icon's down-arrow 
 - Select **Create Variable from Property** in the sub-menu (see Figure 180).
- **Already Set** - If a variable has already been set up, then the full **Set Variables** icon is available.
 - Click the icon.
 - Select another variable via the **Select Variable** dialog. See Figure 181.
 - The **Select Variable** dialog shows the **User Tile Variables** that have been defined for the tile. Select which **User Tile Variable** to use for the selected widget property.
- Click the down-arrow  and select **Configure Variables..** to edit a variable value in the **Variables** dialog or create new, see Figure 181.

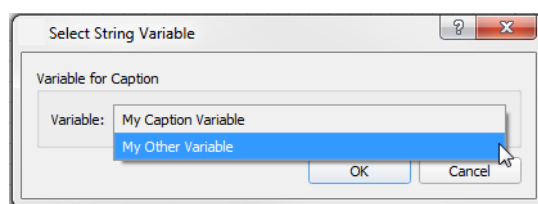


Figure 181 Select Variable Dialog

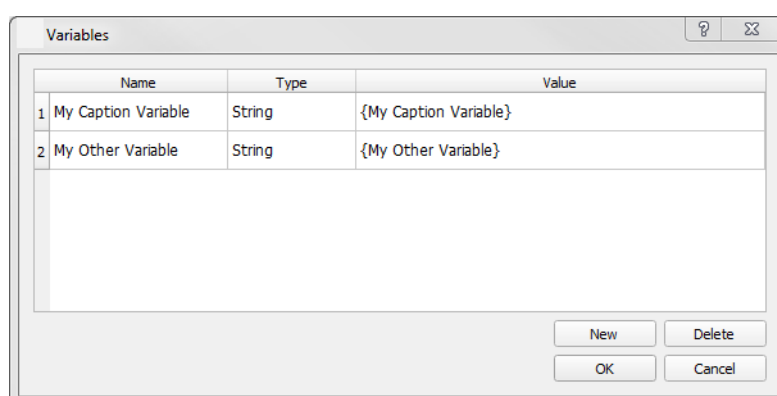


Figure 182 Variable Dialog

The examples that follow illustrate the use of the **Set Variables** function and **User Tile Variables**.

7.2.3 User Tile Variable and Set Variables Example:

1. Create a new empty tile.
2. Add a **Clock** widget onto the tile. See Figure 183.

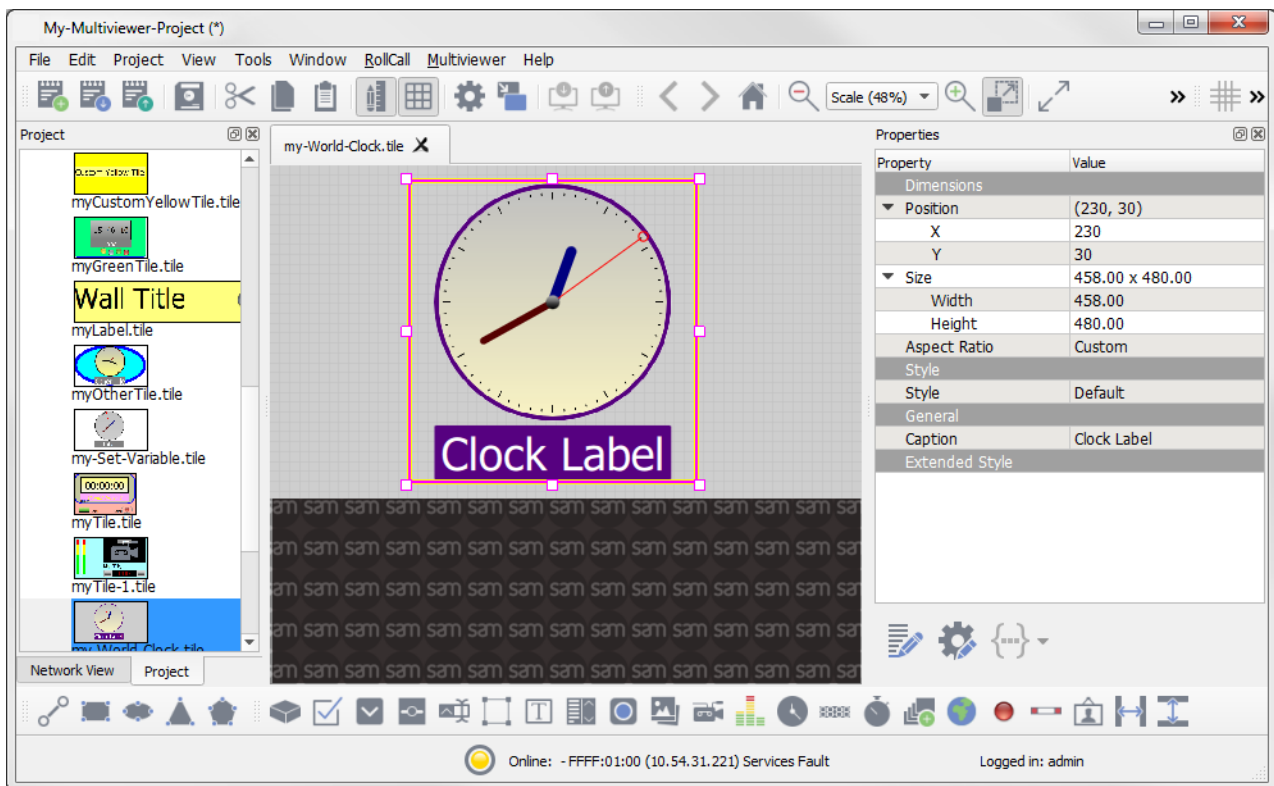


Figure 183 New World Clock Tile

3. *Deselect* all widgets.
On the tile, Right-click and select '**Variables...**'
4. A **Variables** dialog appears. See Figure 184.
5. Create a new **User Tile Variable** in the **Variables** dialog. See Figure 184.
Click **OK**.

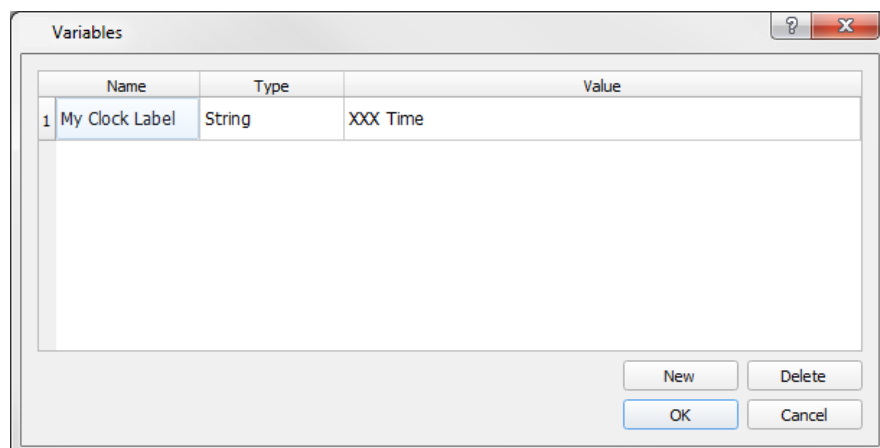
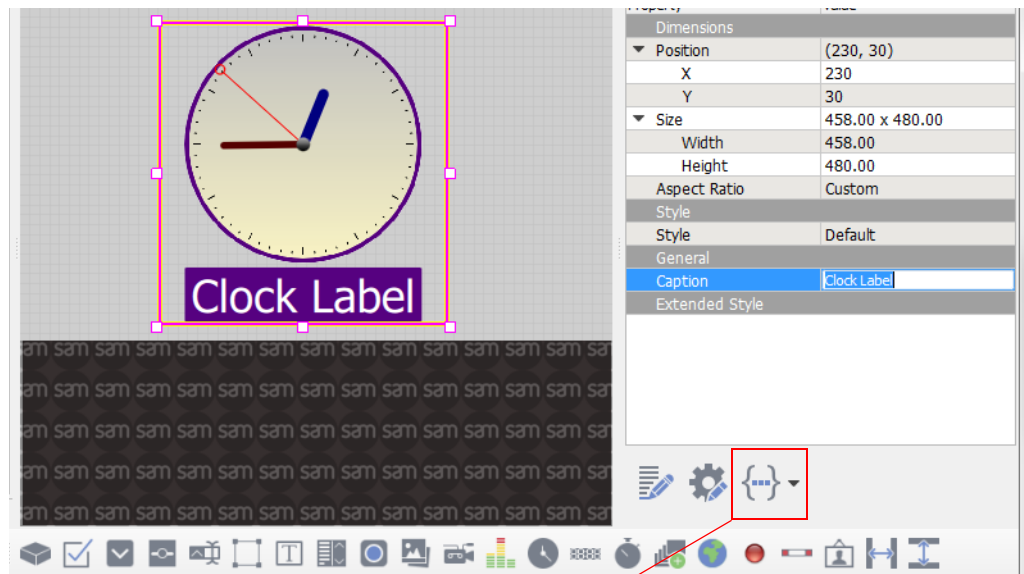


Figure 184 New User Tile Variable, "My Clock Label"

This has created a new **User Tile Variable** for the tile.

6. Select the **Clock** widget.
7. In the properties box, select the widget's **Caption** property.
The **Set Variables** icon in the **Advanced Tool Bar** is now active (not grayed-out).
See Figure 185.



Set Variables icon is active

Figure 185 Clock Widget and Caption Property Selected

8. Click the **Set Variables** icon.
The **Select String Variable** dialog is displayed. See Figure 186.

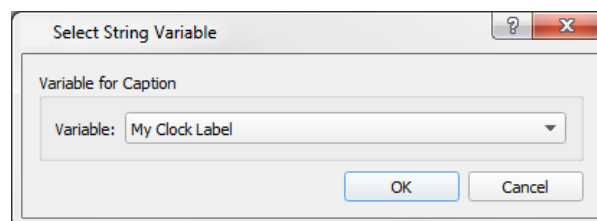


Figure 186 Select Variable Dialog

9. The **User Tile Variable** to use for the widget's **Caption** property can be selected.
In this case, there is only one **User Tile Variable**, so it is a choice of one.
10. Click **OK**.

The **Clock** widget's **Caption** property is now assigned the **User Tile Variable** 'My Clock Label'. See Figure 187.

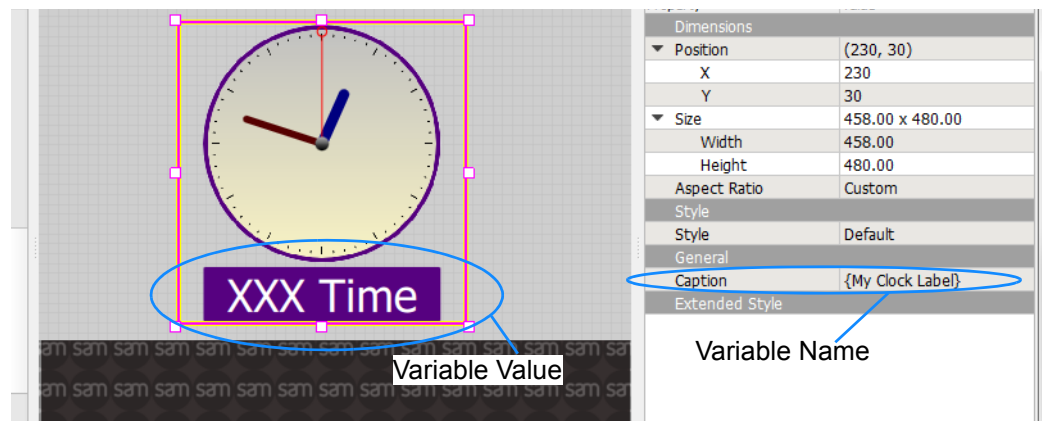


Figure 187 User Tile Variable Assigned

- Click **Save** in the **Main Tool Bar** to save the **Tile**.

The new tile has now been created with a **Clock** widget.

Now we place the **Tile** onto a **Wall**.

- Create a **Wall** or open an existing **Wall**.
Drag some of the new clock tiles onto the **Wall**.
- Select one of the new clock **Tiles** and inspect its properties in the **Properties Box**.
The **User Tile Variable** 'My Clock Label' now appears as a **Tile** property on the **Wall**.
(See property '**Tile > Configuration > My Clock Label**'). See Figure 188.

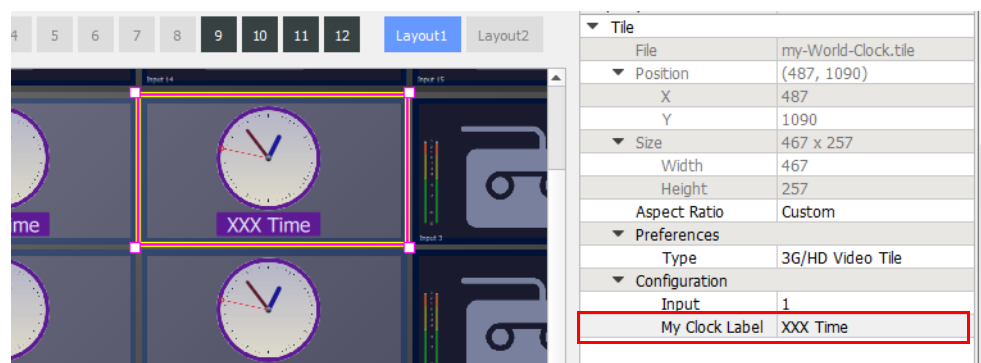


Figure 188 the New Clock Tile on a Wall

- Select the 'My Clock Label' property and change its value, for example, to 'New York', as shown in Figure 188.
The Clock widget label on the **Tile** on the **Wall** now reflects the new property value.
- Select each other new clock **Tile** in turn and change each 'My Clock Label' property.
The Clock widget labels now reflect each **User Tile Variable** that has been defined at **Wall**-level.
See Figure 189.

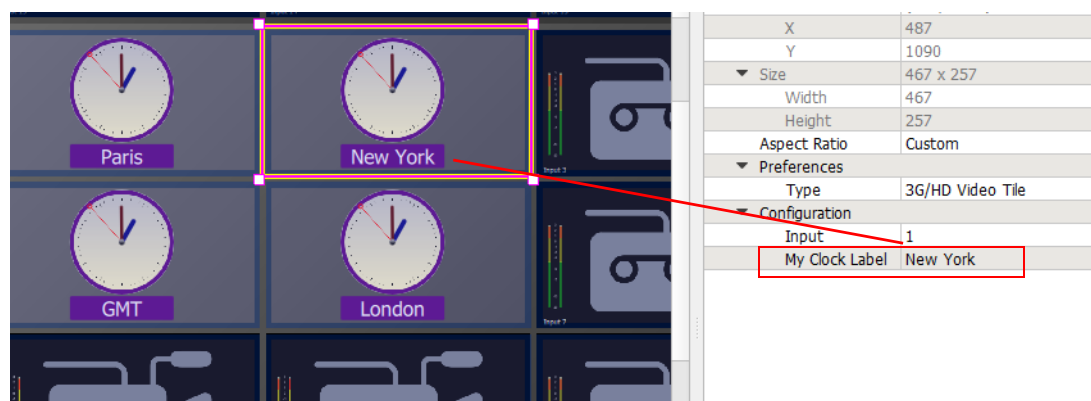


Figure 189 New Clock Tiles on a Wall

A **Tile** has been designed with a parameter (**User Tile Variable**) that can be set at wall level. The **Tile** has been re-used on a **Wall** several times with different **User Tile Variable** values in each case.

In Figure 189, we have four clock faces which do not yet show different world times. The introduction of a further integer type **User Tile Variable** can be used to offset the time shown by each clock. Figure 190 shows four clocks each showing different times, offset by a new **User Tile Variable** 'clock_hour_offset'.

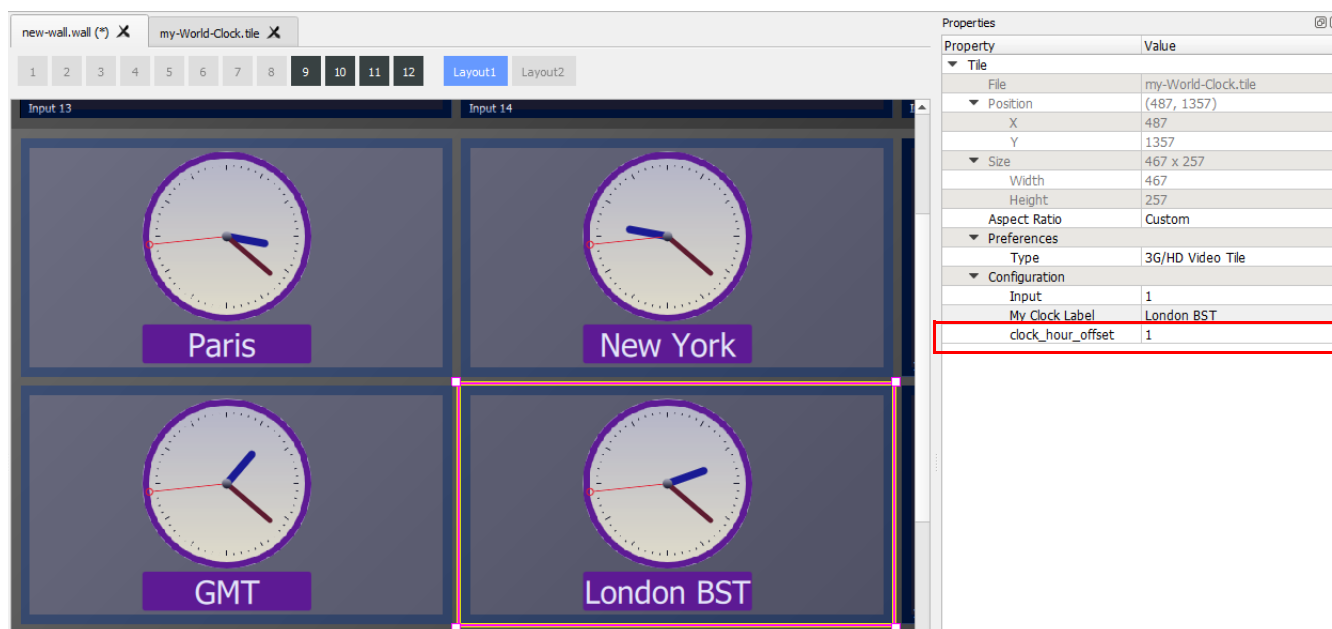


Figure 190 World Clock Tiles on a Wall

7.2.4 'Create Variable from Property' Example

Any property can be set as a variable. This example uses the 'Create Variable from Property' feature with text size on a Label widget.

7.2.4.1 Lines of Text

A Label widget can be configured to add extra lines or text or not. By default a **Label** widget will add extra lines of text if the caption text is too long for the width of the **Label**. See Figure 191b.

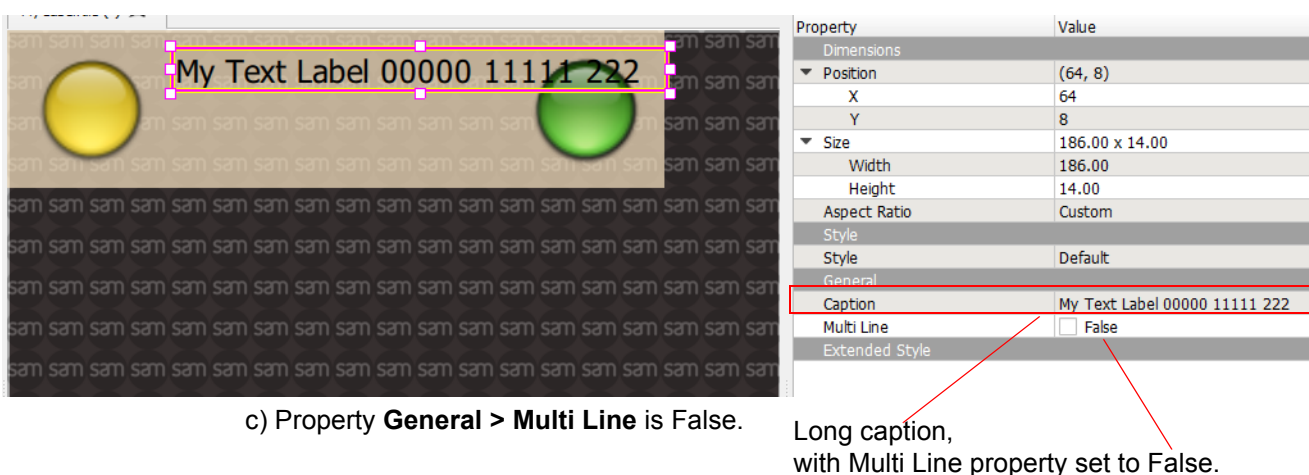
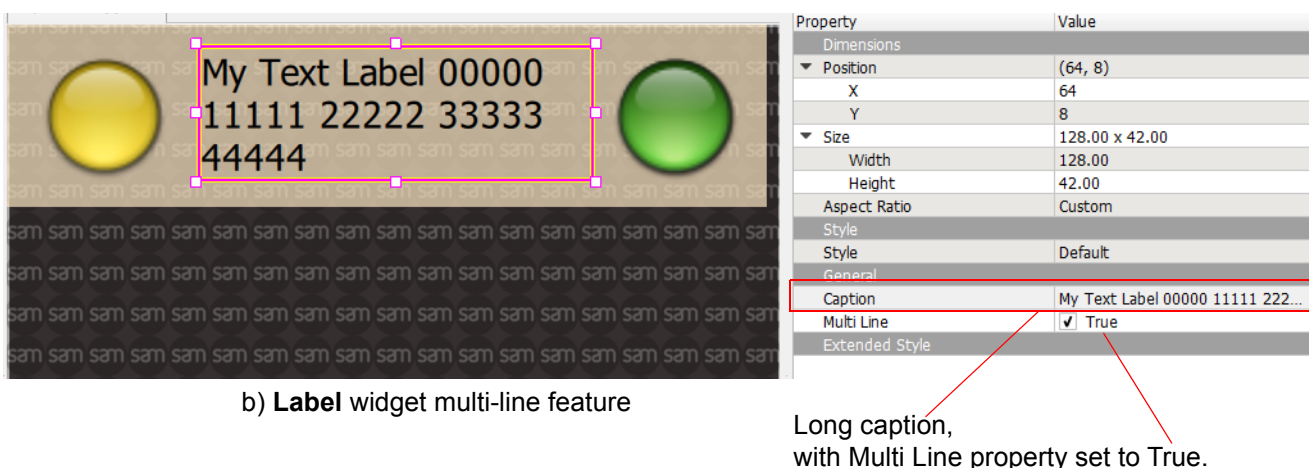
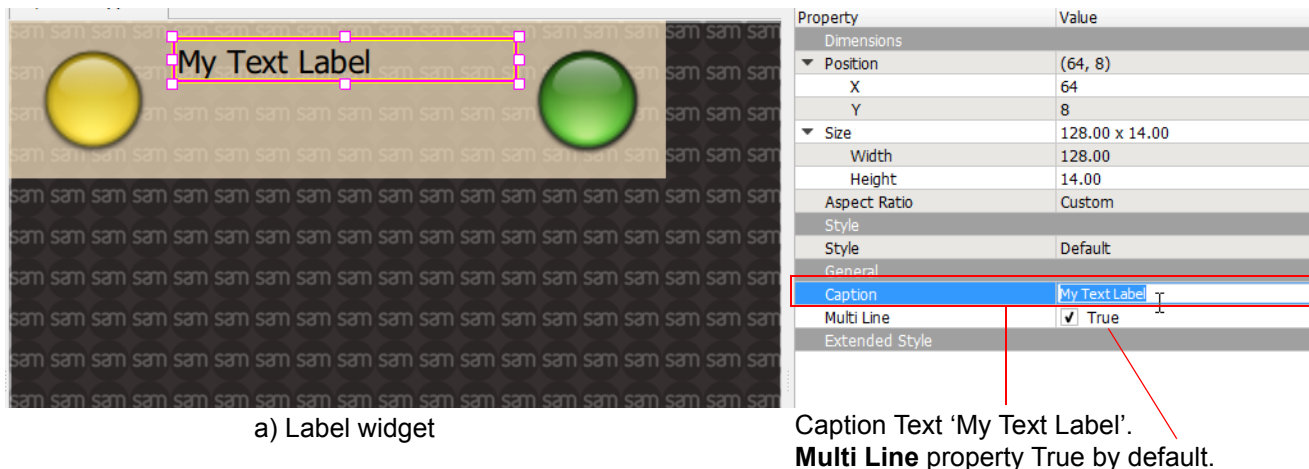


Figure 191 Label and Multi Line Property

7.2.4.2 Text Size

By default a **Label** widget's text size scales with the **Label** widget size (i.e. property **Font > Font Size Mode** is set to 'Variable Size') and the **Font > Font Size** property is grayed out.

Setting property **Font > Font Size Mode** to 'Fixed Size' results in a font size determined by the **Font > Font Size** property, regardless of the label size. This gives manual control over the Label font size on-screen.

See Figure 192.



Figure 192 Label Font Size Mode

The font size can be specified at wall level by setting the **Font Size** property value to a new **User Variable**. The variable can then be set each time the **Tile** is used on a **Wall**:

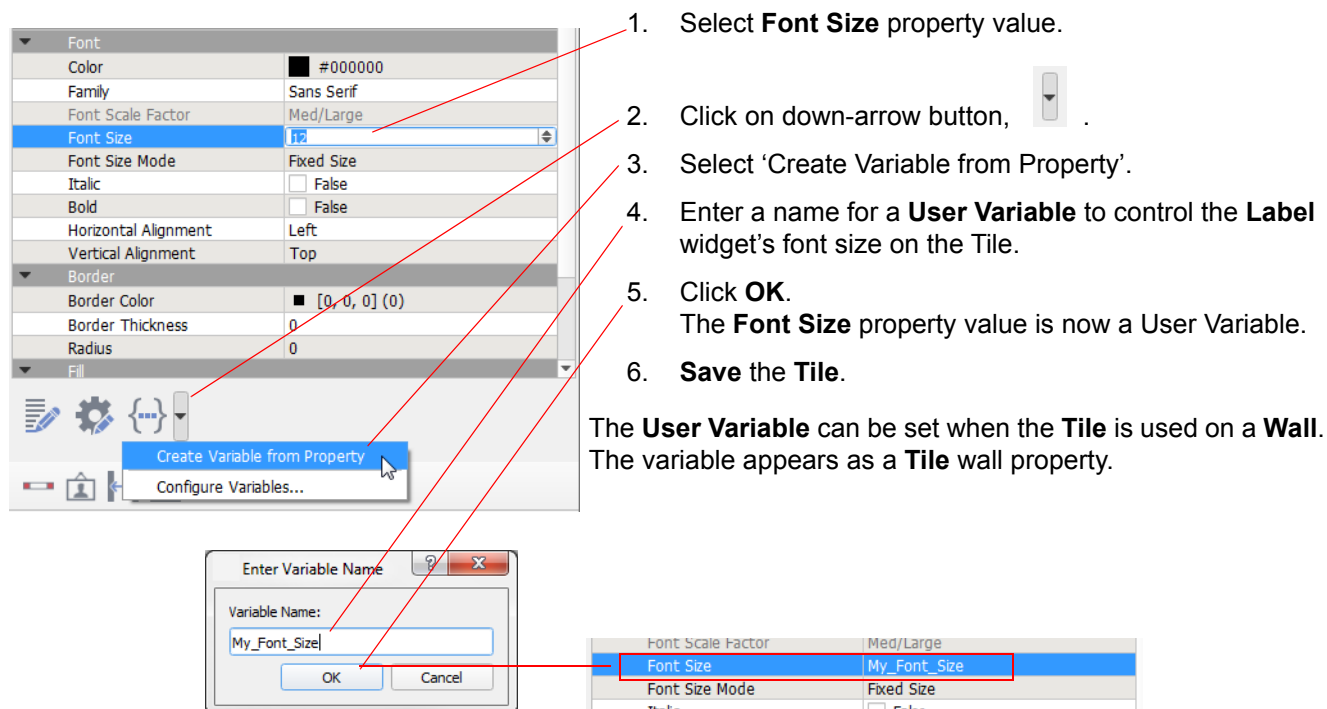
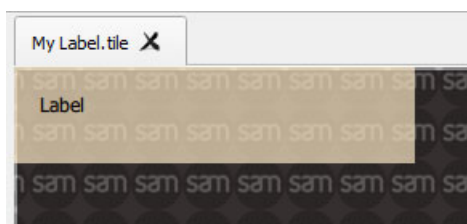
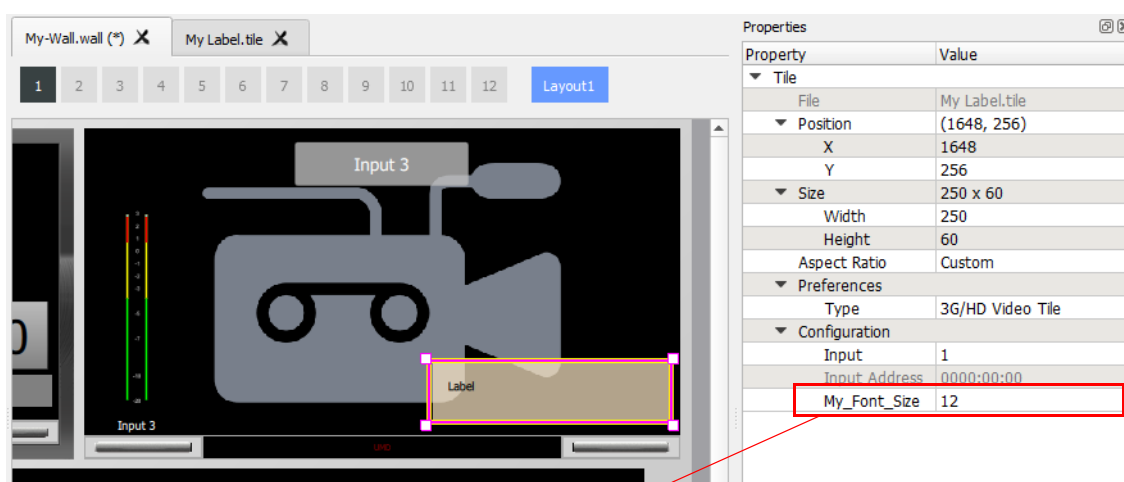


Figure 193 Set Label Font Size to a User Variable

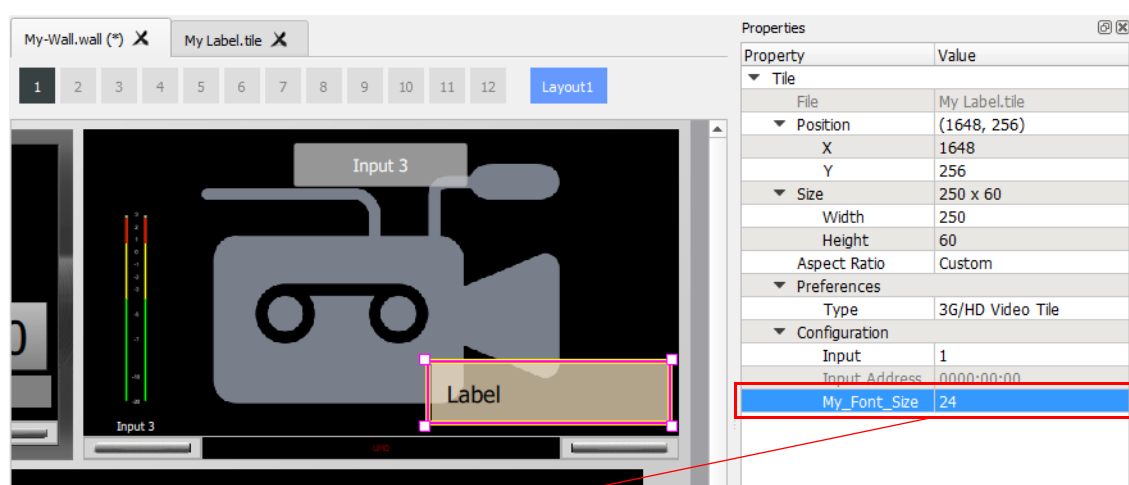
Figure 194b and c show the same **Tile** placed on a **Wall**.



a) Simple **Tile** with a **Label** widget in **Tile Editor**



b) **Tile** on a **Wall**, with the new **My_Font_Size** **Tile** property showing.



c) New **My_Font_Size** property set to 24.

Figure 194 Setting a Tile Property at Wall Level

8 Example - Basic Multiviewer Video Wall

Example - Basic Multiviewer Video Wall

| | |
|---|-----|
| Default Project | 189 |
| Pull Default Project from Multiviewer (Connected Multiviewer Project) | 190 |
| Make a Simple Change to the Default Project | 191 |
| Push the Project to the Multiviewer | 194 |
| Exercising the alarm | 195 |
| Orbit Initial Screen | 196 |
| Creating a New Project | 197 |
| New Project - Multiviewer project | 197 |
| New Project - Connected Multiviewer Project | 200 |
| Opening an Existing Project | 203 |
| Opening a Recent Project | 204 |
| Orbit Project Home Screen | 205 |
| Orbit Project Status Bar | 206 |
| Making a Basic Video Wall Layout | 207 |
| Create a New Multiviewer Project | 207 |
| Create the Video Wall | 209 |
| Import a Logo Image into the Project | 213 |
| Create Custom Tiles | 214 |
| Add Custom Tiles to the Wall | 216 |
| Assign Multiviewer Inputs to Wall Video Tiles | 218 |
| Push Project to the Multiviewer | 220 |
| Before Pushing a Project | 220 |
| Check Project Name | 220 |
| Push the Project to the Multiviewer | 220 |
| Wall Layouts | 221 |
| Themes | 223 |
| Baseline Default Theme | 223 |
| Creating Themes | 223 |
| Applying Themes to a Wall | 225 |
| Applying a Style to a Widget | 226 |
| Locally Modifying the Style of a Widget | 226 |
| Default Style | 227 |

This section contains sub-sections describing some basic changes to a video wall.

Multiviewer monitor wall layouts are designed with the Orbit application and are called 'Orbit projects'. These multiviewer projects can be pushed to a multiviewer to be deployed and the video wall can be viewed on monitor displays. Additionally, video wall layouts can be pulled from a multiviewer device and modified. Also, graphical control and monitoring screens can be designed, and the form and appearance of a wall design may be changed through layouts and themes.

8.1 Default Project

A Orbit multiviewer project contains all the information for a multiviewer video wall layout and a new Grass Valley MV-8 series multiviewer device contains a default multiviewer layout, which comprises:

- Four video walls of 3x4 video pictures, one per multiviewer display output.

Figure 195 shows the appearance of each of the four multiviewer display outputs, Output 1 to 4, and the assignment of multiviewer inputs across these output screens.

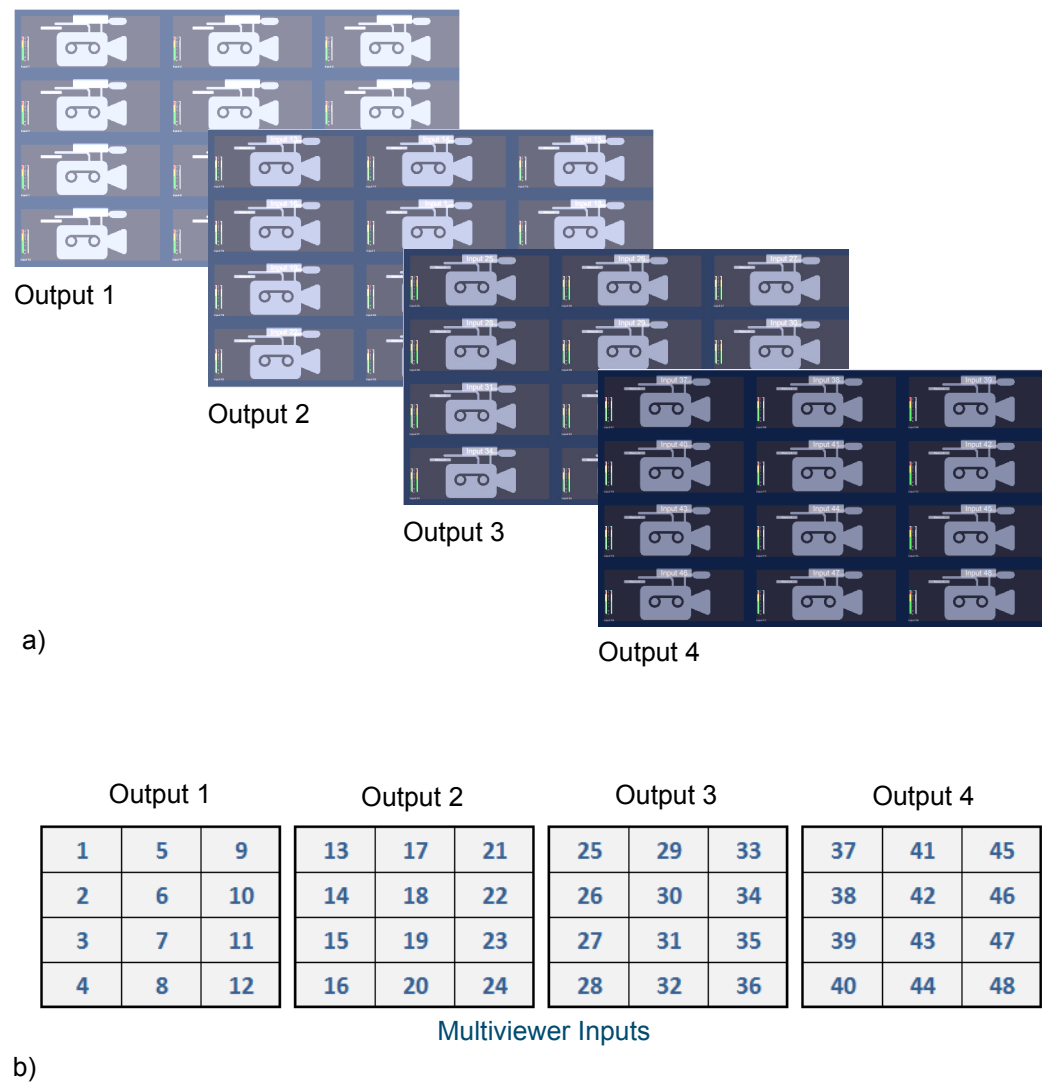


Figure 195 Grass Valley MV-8 Series Multiviewer Default Video Wall:
a) Assignment of Multiviewer Outputs
b) Assignment of Multiviewer Inputs 1 to 48 across Multiviewer Outputs 1 to 4.

Layouts stored on a Grass Valley MV-8 series multiviewer may be read as a multiviewer project by Orbit. The default Orbit project login user name and password are both ‘admin’.

8.1.1 Pull Default Project from Multiviewer (Connected Multiviewer Project)

The Orbit multiviewer project on a new Grass Valley MV-8 series multiviewer is the default multiviewer project.

Note:

If a project is new to the instance of the Orbit software being run, then Orbit considers the project to be a 'new project' and a new Orbit project needs to be created.

Creating a new project from an Orbit for Multiviewers which Orbit is connected to is called 'creating a Connected Multiviewer Project'.

Firstly, we pull the default multiviewer information (Orbit project) from the multiviewer:

1. Run the Orbit software application on a PC.
2. Click **File > New Project**.
Click on the **Connected Multiviewer Project** icon.
3. **Browse** to a PC location and into an empty PC sub-folder, where you wish information read from the multiviewer to be stored on the PC.
Note: *The sub-folder must be empty.*
4. Go *into* the empty sub-folder and then click **Choose**.
Click **Next**.
A list box for available multiviewer units appears.

Note:

The empty sub-folder name will be used as the project name of the pulled project.

5. Select the unit from the list of discovered units.

Note: If multicast discovery is not available, a RollCall Domain ID and unit IP Address can be specified after clicking **Specify Host**.

Click on a **unit name**, to select the unit.
Click **OK**.

6. Use "**admin**" as user name and password.
Click **Login**.

The multiviewer's video wall layout is pulled from the multiviewer, stored in the specified PC sub-folder and displayed in Orbit.

The Orbit Project Home screen is shown, see Figure 196.

7. Change the **Project Name** on the Orbit Project Home screen by clicking on **Edit Name**.
(Re-name a default project and use unique project names for each multiviewer.)

Click **Edit Name** to change the Project Name

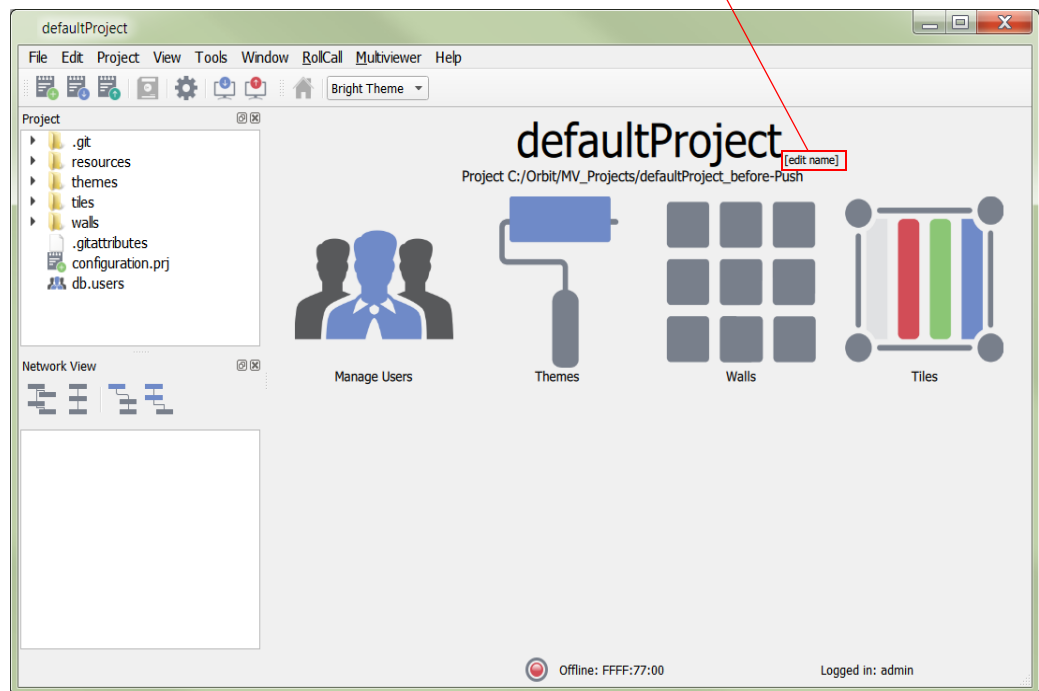


Figure 196 Orbit Project Home Screen

Note:

A blank new project may be created from scratch by:

Clicking on the **Multiviewer Project** icon instead of **Connected Multiviewer Project** icon.

This creates a blank new project instead of a project from information pulled from a multiviewer. See Section 1.3 “Setting up Multiviewer Device Details in a Project” on page 4 for setting multiviewer device details in a project.

Use a unique project name for each multiviewer.

8.1.2 Make a Simple Change to the Default Project

Next, we shall make a simple visible change to the project and enable an alarm across all multiviewer inputs:

1. Open one wall of the project by doing one of the following:
 - a) Click the **Walls** icon and click on a wall (for example, “**Wall1**”) in the drop-down list that appears.

Or

 - b) In the **Project View**, on the left-hand side of the screen:
 Expand the **walls** sub-folder.
 Right-click on a **Wall** file name (for example, ‘Wall1.wall’)
 and click **Open** in the selection menu that appears.
 Or simply double-click on the **Wall** file name.

The **Wall Editor** screen appears, showing the wall on a multiviewer display output.

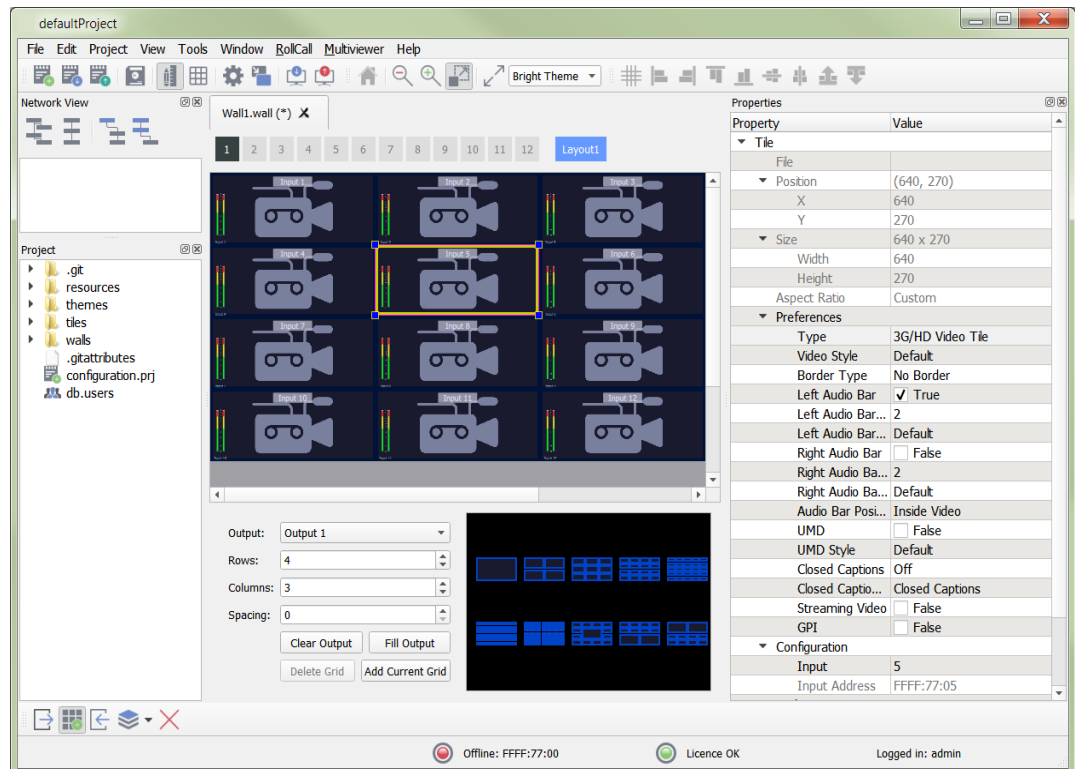


Figure 197 Orbit Wall Editor Screen - Default Wall

Note:

The relative sizes of the various Orbit screen areas and panes (**Network View** pane, **Project View** pane, **Properties Box** and central schematic area) may be changed by dragging the boundaries with the cursor.

Next:

2. Click on one of the middle tiles in the wall.
The **Properties** box, on the right-hand side, shows the selected tile's properties.
3. Change the tile **Type** to 'Analogue Clock'.
The selected tile changes to a round clock face.



4. Click the **Save File** icon to save the change.

A visible change has now been made to the wall design in the Orbit multiviewer project, see Figure 198.

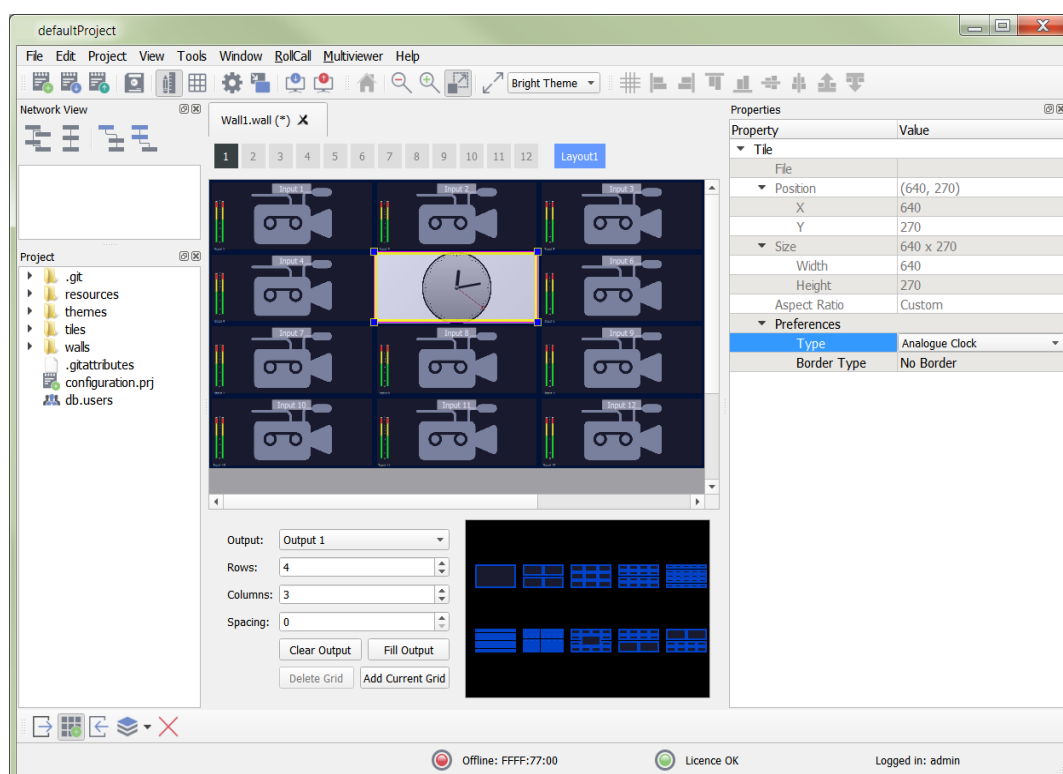


Figure 198 Orbit Wall Editor Screen showing a Clock Tile

Now proceed to configure simple alarms for each multiviewer input:

5. Click **Multiviewer > Alarms**.
An **Alarms Dialog** is shown, with some tabs.
6. Select the **Input** tab:
Set **Selected Input** to **Input 1**.

7. Select the **Alarm** tab.
Set **Selected Alarm** to **Video Input Lost** (scroll down the list of alarms).
Select **Alarm Enable**. See Figure 199.

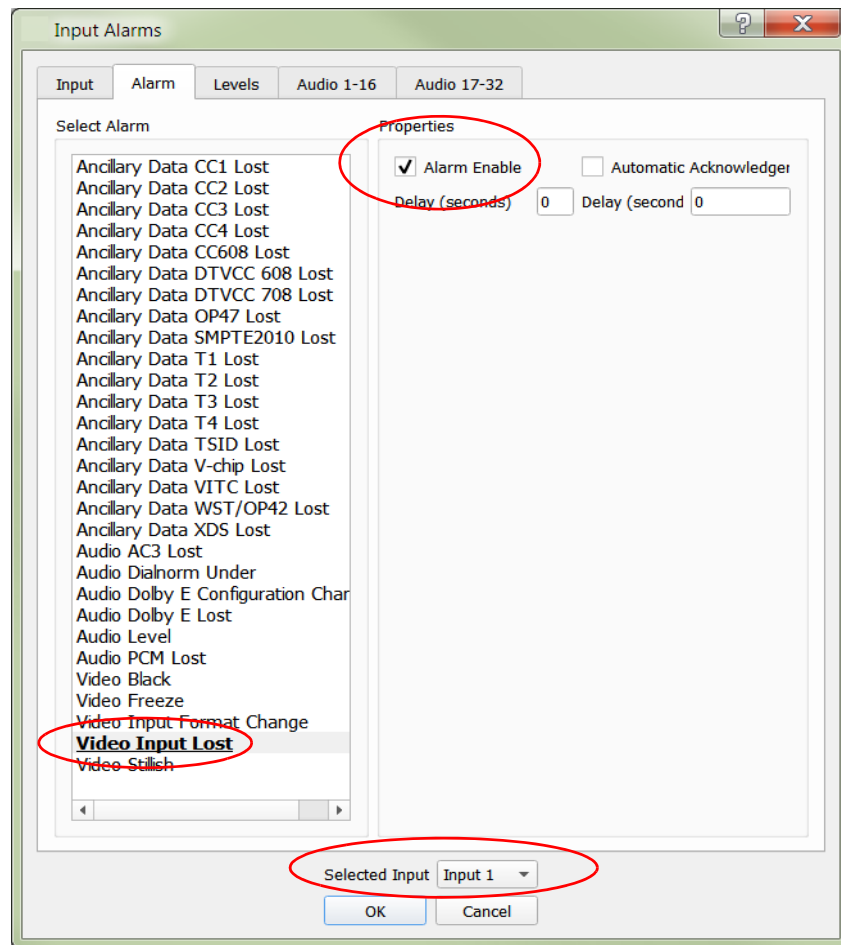


Figure 199 Alarms Dialog - Video Input Lost Alarm is Enabled

8. Select the **Input** tab again:
Click **Copy All**.
This copies the alarm settings of the **Selected Input** (here **Input 1**) to all inputs.



9. Click the **Save File** icon to save the change.

Video Input Lost alarm is now enabled for all multiviewer inputs.

8.1.3 Push the Project to the Multiviewer

Now we push the changed project back to the Multiviewer:

10. Check that the project has a unique Project Name; re-name the project if required.
11. Click **Project > Select Multiviewer**.
Enter the **IP address** of the Multiviewer.
12. Click **Project > Push**.
The modified project is pushed to the multiviewer.

The multiviewer adopts the new wall layout. For this simple change, an analogue clock appears on an output and one alarm is enabled on each multiviewer input.

8.1.4 Exercising the alarm

The Video Input Lost alarm may be triggered by routing an unconnected router video source to a multiviewer input.

13. Route an unconnected router video source to multiviewer **input 1**.

When the alarm is triggered:

14. A **flashing red border** appears around the corresponding video tile on the wall.
(**Input 1** is the top-left tile of Output 1)



Figure 200 Alarm Red Border around a Video Tile

A simple check to verify Orbit is registering alarm messages from the multiviewer is to:

15. Expand the multiviewer name in the **Network View Pane**, left-hand side of the Orbit screen.
Right-click on the '**01: Input 1**' item and select **Details**.
A **Details** text window appears, showing the status of Input 1. See Figure 201.

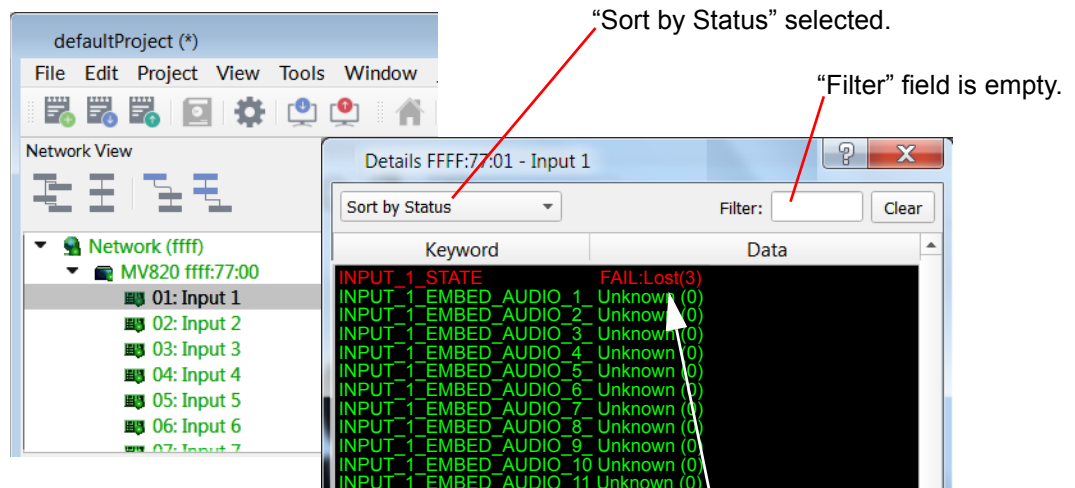


Figure 201 Input 1 Details Text Window, showing **RED** "Lost" Alarm Message text.

16. A red text "**Lost**" text message appears for Input 1, indicating a triggered alarm.

Details:

Status details are continuously updated in the **Details** text window.

Changed status values are highlighted in the **Details** window with an **orange background** for a few seconds.

8.2 Orbit Initial Screen

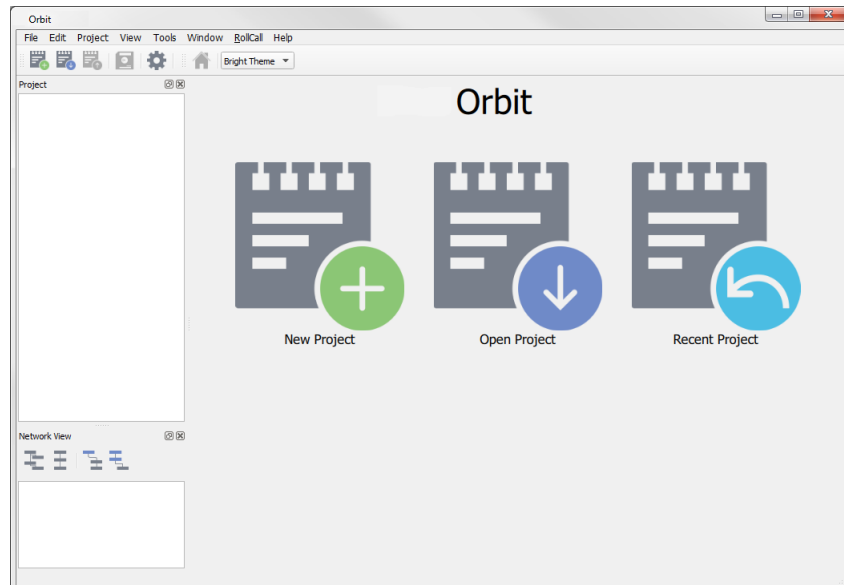


Figure 202 Orbit Initial Screen

Orbit stores multiviewer configurations as Orbit projects. The icons on the Orbit Initial Screen allow the user to access a project.

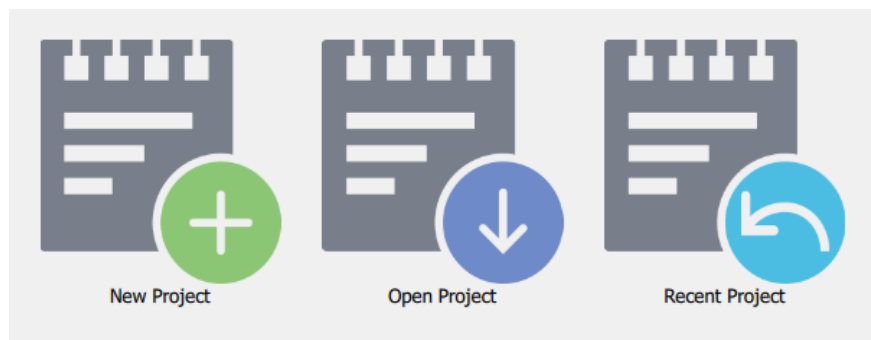


Figure 203 Orbit Initial Screen Icons

Note: The **Recent Project** icon is only displayed if one or more projects have already been created by the Orbit installation.

The icons on the Orbit Initial Screen enable a user to open a project. They are:

- **New Project** - Opens the New Project dialog box. A new project can be created from scratch or a project can be obtained from a unit that Orbit is connected to. For example, a multiviewer unit.
 - See sub-section about “Creating a New Project” on page 197.
- **Open Project** - Opens the Open Project dialog box. The user can open an existing Orbit project - one which has been saved to storage on a PC or server.
 - See sub-section about “Opening an Existing Project” on page 203.
- **Recent Project** - Reveals a drop-down list of projects that have been recently accessed by your Orbit application. Project names are listed.
 - See sub-section about “Opening a Recent Project” on page 204

Once a project is open, the Orbit Project Home screen is displayed with four large icons showing, see Section 8.6 “Orbit Project Home Screen” on page 205

8.3 Creating a New Project

There are two types of new project that may be created:

- **Multiviewer Project** - Creates a brand new project on the local PC for an Orbit multiviewer for editing in the Orbit application.
- **Connected Multiviewer Project** - This option connects to a multiviewer and pulls the project from the unit onto the local PC for editing in the Orbit application.

8.3.1 New Project - Multiviewer project

1. Click on the **New Project** icon on the Initial Screen or select **File > New Project** from the menu bar.
2. Select the **Multiviewer Project** icon to create a new project.

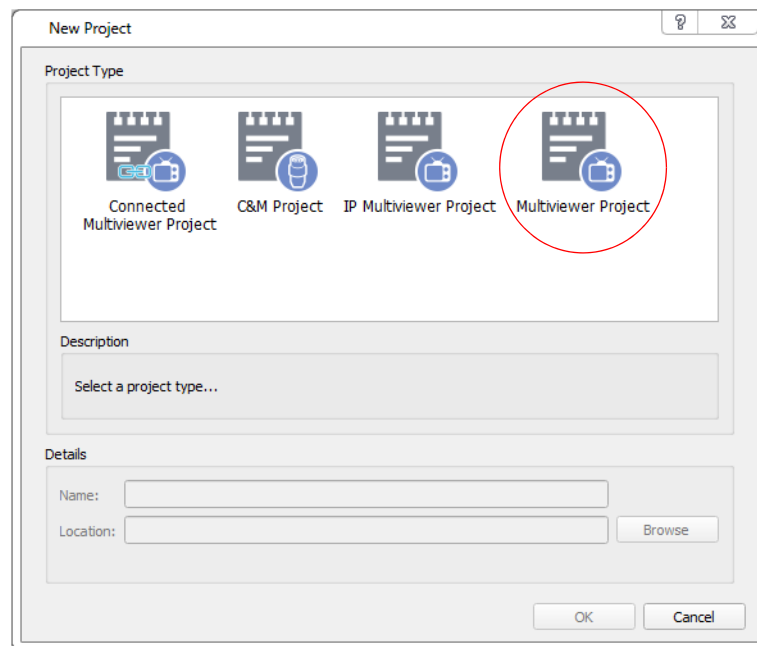


Figure 204 New Project window

3. Enter a name for the project in the **Name** text box. Use a unique name for each multiviewer.

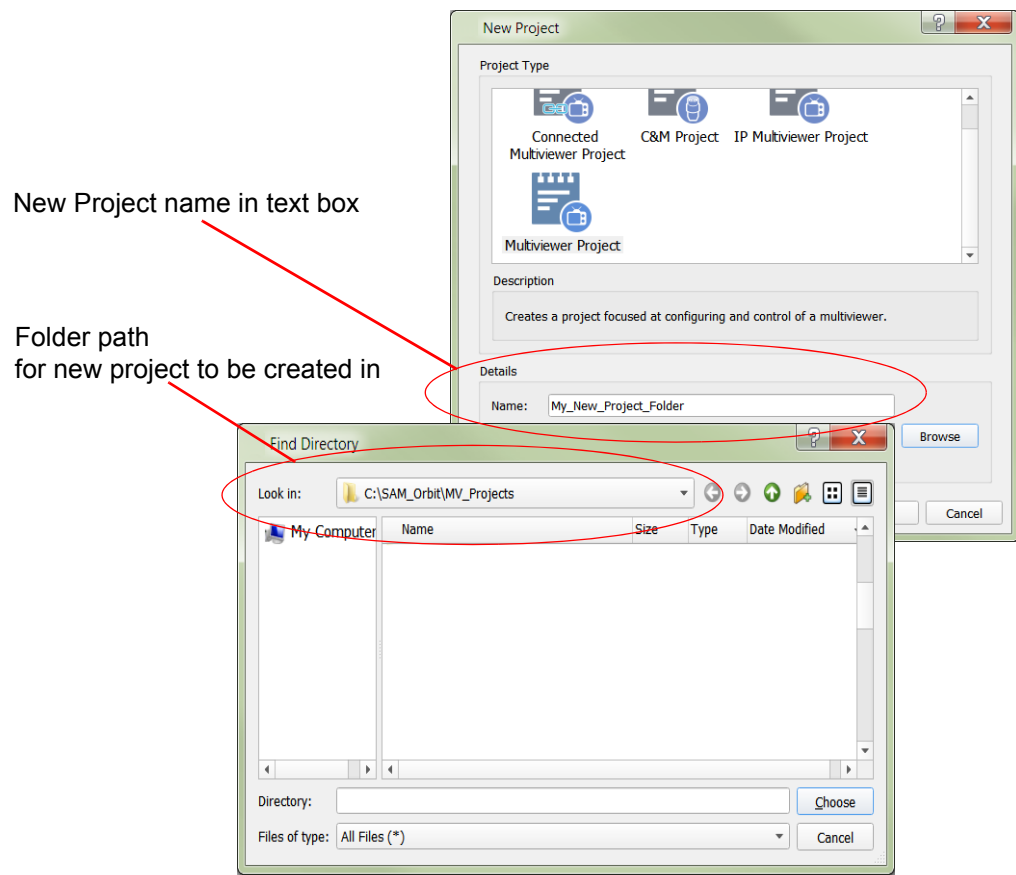


Figure 205 Creating a Multiviewer Project - Name text box, Find Directory window.

Note:

Project Name:

The project name entered in the **Name** text box is used to identify the project. Project names should be meaningful, concise and unique. Re-name any default project that has been pulled from a multiviewer.

Note:

Orbit Opening a Project:

When opening a project, Orbit will:

- Try and connect to the specified Multiviewer.
- Check for a project with the same name.
- Check for any differences between the local project and the project on the Multiviewer device.
- Warn the user if the two projects have the same name but are different.
- Prompt the user to decide which copy of a project is to be edited in Orbit.

4. The new project will be stored in a folder. **Browse** down *into* the main folder that the new project is to be stored in using the **Find Directory** window. See Figure 205.

5. When in the desired folder, click **Choose**.
The **New Project** window is shown. See Figure 206:

- Your project name should be present in the 'Name:' text box.
- The path to the folder in which you wish your new project to be created should be in the 'Location:' text box.

Note:

When clicking **Choose**, the path to the folder (directory) and the folder name should be in the **Look In:** text box at the top of the **Find Directory** window. See Figure 205.

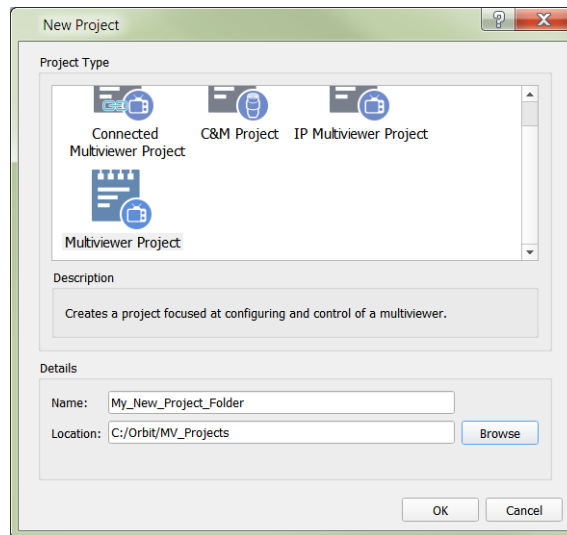


Figure 206 Multiviewer Project Creating - After clicking on Choose.

6. Click **OK**.
The Orbit project is then created. It is actually stored in a sub-folder which takes its name from the project name.
Figure 207 shows the PC folder structure that results.

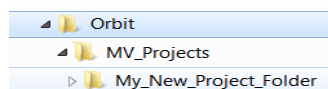


Figure 207 Folder structure on PC

7. By default, you are automatically logged into the Orbit project at 'admin' user level.

Default user name - admin

Default password - admin

Note:

The 'admin' user has administrator role privileges, which enable all operations within Orbit to be carried out.

Important:

There is no user recovery mechanism for the admin password. Keep a careful and secure note of the password used.

A new Orbit project is now open and Orbit Project Home screen is displayed with four large icons showing.

The project does not yet contain details about any multiviewer unit. Details need setting up before the project can be pushed to a multiviewer. See Section 8.8.1 "Before Pushing a Project" on page 220.

8.3.2 New Project - Connected Multiviewer Project

If an MV-8 series multiviewer unit contains an existing Orbit multiviewer project, it can be pulled off the unit into Orbit, enabling the user to store the project in a selected folder on their PC and edit the project:

1. Click on the **New Project** icon from the Home Screen or select **File > New Project** on the menu bar.
2. Select the **Connected Multiviewer Project** icon to create a new project with.

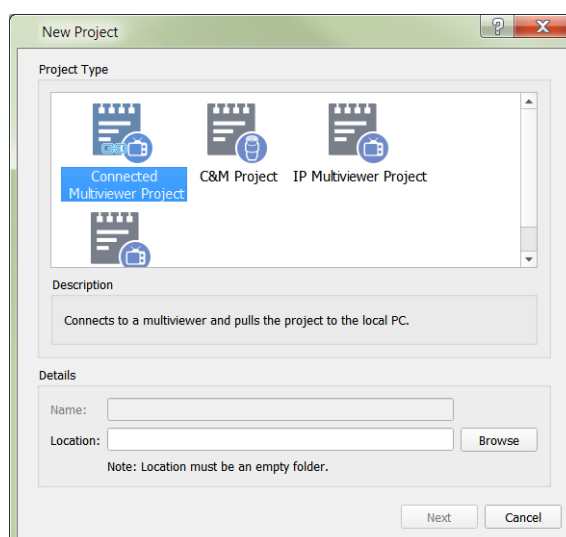


Figure 208 Selecting Connected Multiviewer Project

3. Click **Browse** and browse *into* an *empty* folder on your PC, where an Orbit project can be placed after it has been pulled from a multiviewer unit. Click **Choose** when you have browsed into the folder, see Figure 209.

Note:

When choosing a folder for a Connected Multiviewer Project to be pulled into, the folder must be *empty*.

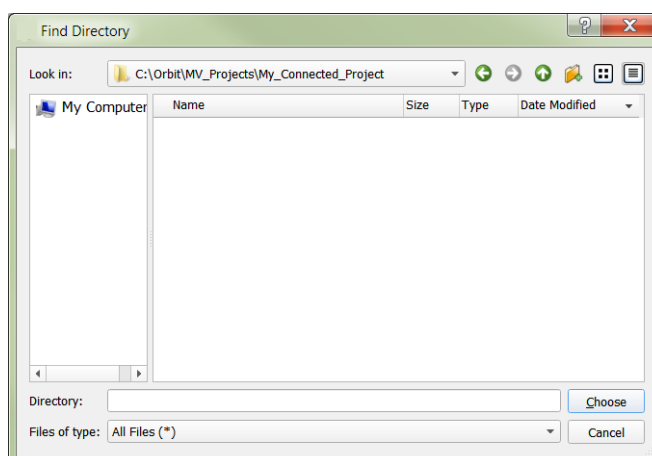


Figure 209 Choose empty folder for Connected Orbit project

- After Choosing the folder location, click **Next**.

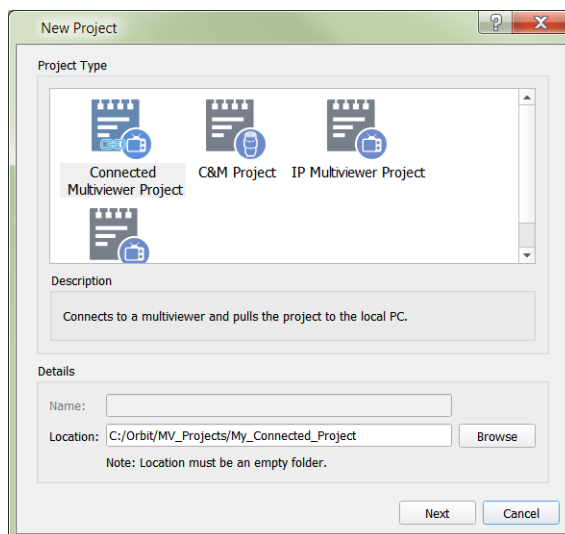


Figure 210 Folder Chosen for Connected Orbit project

- Select your multiviewer unit from the ones listed. Check that the Domain number is correct for your units. Click **OK**.

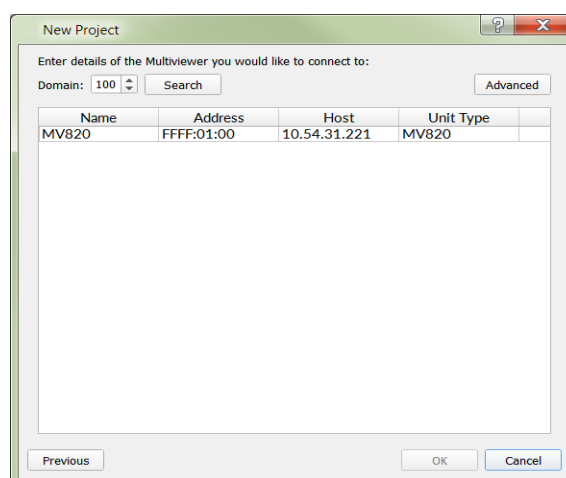


Figure 211 Selecting Multiviewer unit

- The Orbit project is pulled from the multiviewer and placed in the selected folder.

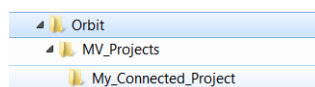


Figure 212 Folder structure for a Connected Project pulled onto a PC

7. You are prompted to log into the connected Orbit project, because users and passwords will have been set up for the connected project at creation.

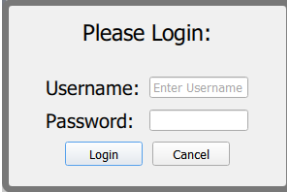


Figure 213 Folder structure for a Connected Project pulled onto a PC

8. To log in:
 - Enter your user name in the **Username** text box.
 - Enter your password in the **Password** text box.
 - Click **Login** to log in, or click **Cancel** to cancel the log in.

Note: Generally, when opening a new connected project, you must be a user of that connected project to be able to log in.

Note: Default projects that ship with Orbit multiviewer products have default admin/admin user/password.

A new connected Orbit project is now open and the Orbit Project Home screen is displayed with four large icons (see section 8.6).

Details about the connected multiviewer are filled out in the project.

Note: If the project is to be used on a different multiviewer, then change the project name. Use a unique project name for each multiviewer.

8.4 Opening an Existing Project

Note:

The **Recent Projects** icon is only displayed on the Orbit Initial Screen if one or more projects have already been created or opened.

From the Orbit Initial Screen:

1. Click on the **Open Project** icon or select **File > Open Project** on the menu bar.
2. Browse *into* the folder containing the Orbit project to be opened.
3. Select the project file (extension “.prj”) and click on the **Open** button.

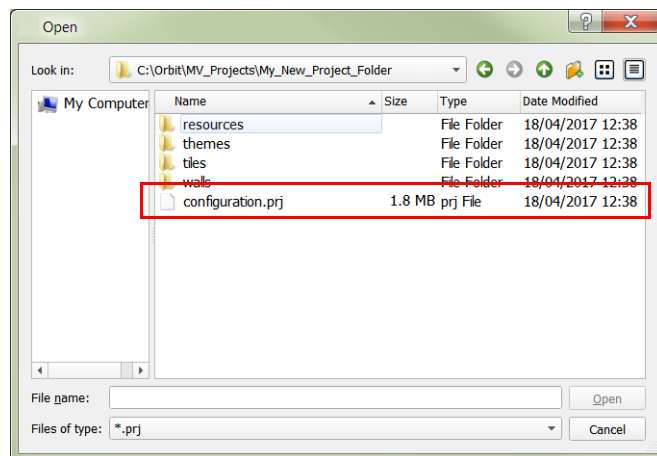


Figure 214 Open Project - Select .prj File

4. A Login dialog box is presented. Enter the username and password when requested.

Default username - admin

Default password - admin

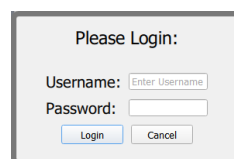


Figure 215 Open Project - Login

5. Click **Login**.
A new Orbit project is now open and the Orbit Project Home screen is displayed, with four large icons.

8.5 Opening a Recent Project

Note:

The **Recent Projects** icon is only displayed if one or more projects have already been created by the Orbit tool installed.

From the Orbit Initial Screen:

1. Click the **Recent Project** icon, or select **File > Recent Projects** on the menu bar.
2. Browse the 'recent projects' drop-down list.
Hover your cursor over a name in the list to reveal the path to that project.

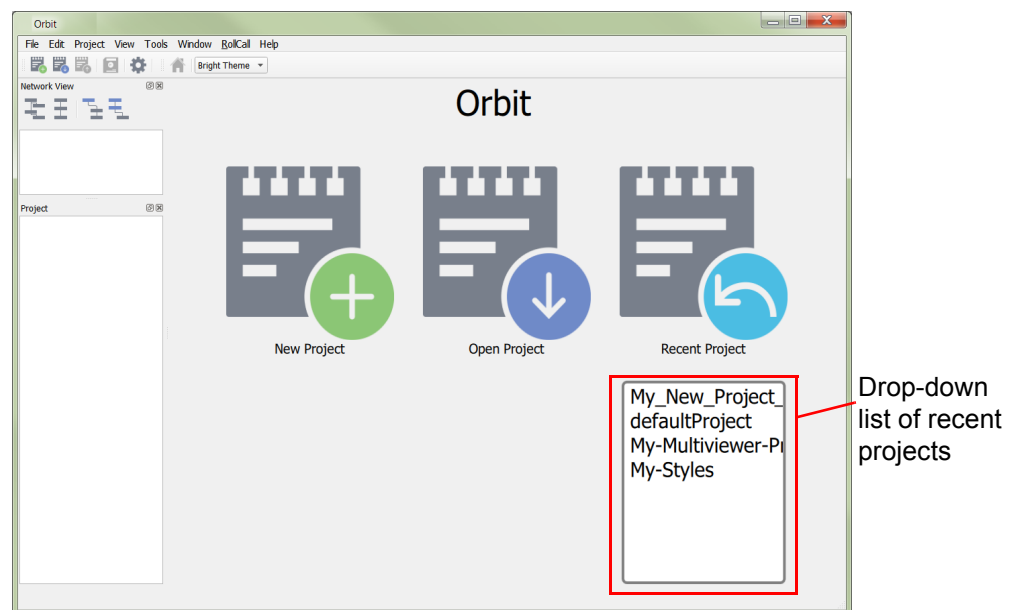


Figure 216 Orbit Recent Projects List

3. Click on the project to be opened in the drop-down list.
4. Enter the username and password at the Login dialog box.
Default username - admin
Default password - admin
5. Click on **Login**.
A new Orbit project is now open and the Orbit Project Home screen is displayed with four large icons.

Orbit stores unit configurations as Orbit projects, allowing various product unit configurations to be viewed, edited and saved.

It is possible to change a product unit's configuration by pushing different projects to it.

The Orbit application's Project Screen (see section 8.6) is displayed upon accessing a project from the Orbit Initial Screen.

8.6 Orbit Project Home Screen

The Orbit Project Home Screen is displayed when a new project has been started or when an existing project is opened.

The Project Home screen is the main hub that gives access to all areas of the Orbit project design.

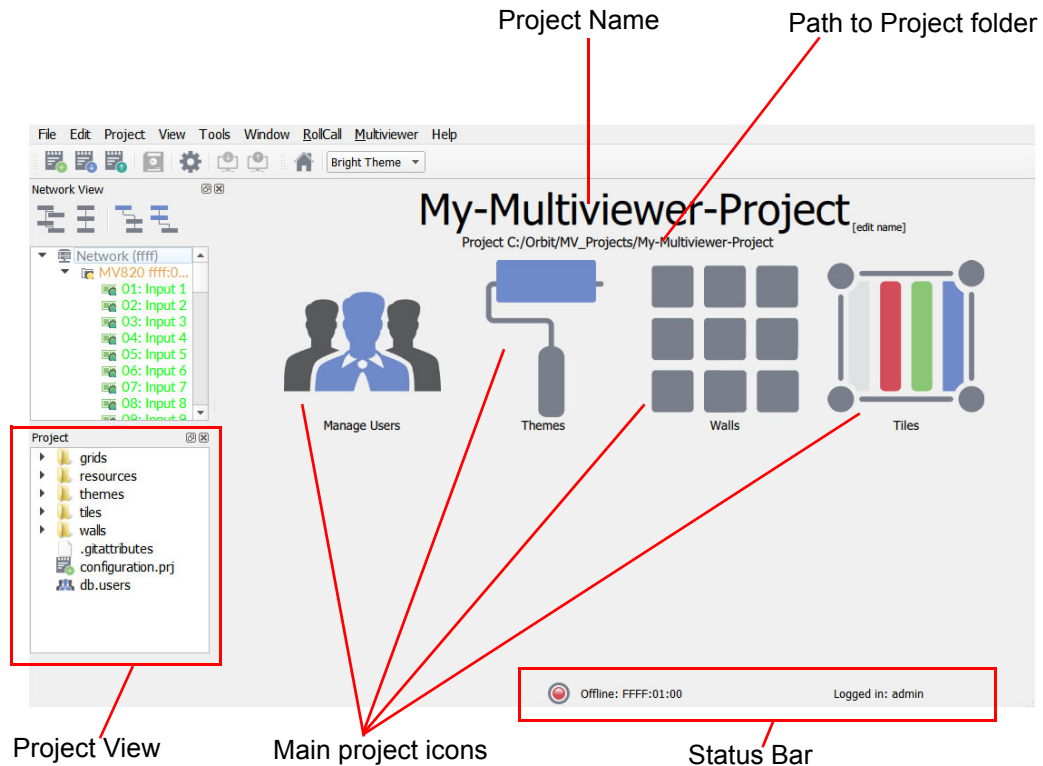


Figure 217 Orbit Project Home Screen

The contents of the project are presented in tree view in the **Project View**.

The main project icons are:

- **Manage Users** Click to manage various levels of user access to the multiviewer project, depending on a user's role.
- **Themes** Click to design widget styles to be used on multiviewer video walls and tiles.
- **Walls** Click to configure the multiviewer wall design and layout.
- **Tiles** Click to create tiles that can then be added to the multiviewer video wall design.

The Status Bar shows status information while an Orbit project is open, see Section 8.6.1 Orbit Project Status Bar.

8.6.1 Orbit Project Status Bar

Information is shown in the Status Bar at the bottom of an Orbit screen when an Orbit project is open. There are up to three status fields shown.

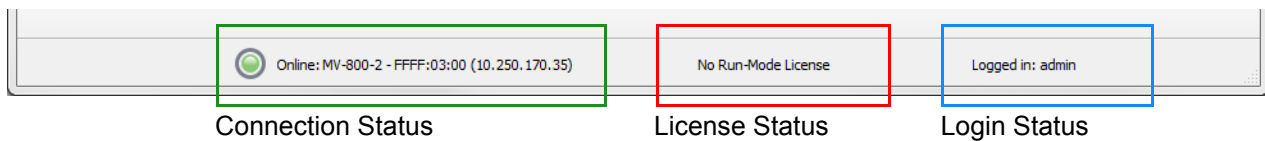


Figure 218 Orbit Status Bar Status Fields

| Status Fields | Description |
|---|--|
| Connection Status: | |
| LED: | |
| Green | Orbit connection to multiviewer unit is OK. |
| Amber | Multiviewer present, but connection not fully working. |
| Red | Multiviewer is off-line for Orbit. |
| Note: It may still be possible to open and view a RollCall template from the Multiviewer. | |
| Online/Offline: | |
| Online | Orbit has connection to a multiviewer unit's services: |
| Offline | No Orbit connection to any multiviewer services. |
| <unit name> | The multiviewer unit's name is displayed. |
| FFFF:01:00 | Unit's RollCall address. |
| (<IP Address>) | Unit's IP address. |
| Services Fault | One or more of the multiviewer unit's services are missing. |
| Possibly reasons for this: | |
| <ul style="list-style-type: none"> Network settings preventing multiviewer discovery. Multiviewer configuration is incorrect. For example, the multiviewer's network interface for RollCall to use is wrongly set up in the multiviewer's RollCall template | |
| License Status: | |
| Status text field is blank when a license has been successfully acquired or when no license is needed. | |
| No Run-Mode License | Licensing is not enabled (see Orbit Tools > Options > Licensing), and user is in a part of Orbit requiring a license. |
| No Run-Mode License | Licensing is enabled (see Orbit Tools > Options > Licensing), and user is in a part of Orbit requiring a license, but no license is available. |
| Login Status: | |
| Logged in: | Logged in username is shown. |

Table 27 Status Bar Status Fields

8.7 Making a Basic Video Wall Layout

This section outlines how to create a basic video wall layout in Orbit and assumes some prior knowledge of the Orbit application. The section aims to exercise several wall layout features of a Orbit multiviewer project.

8.7.1 Create a New Multiviewer Project

1. Click the **New Project** icon.
2. Click the **Multiviewer Project** icon.
Enter a **Name** for the project.

Note:

Use a unique project name for each multiviewer.

3. **Browse** to a PC folder location where the new project will be stored.
Click **OK**.

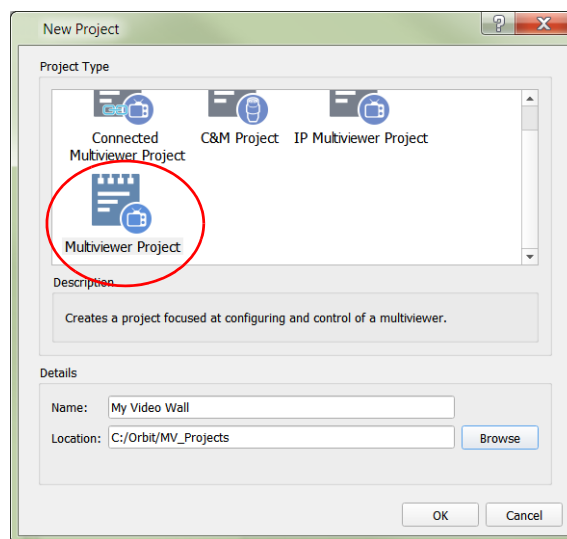


Figure 219 Orbit New Project Dialog

A new Orbit multiviewer project (an unconnected project) is created at the project location specified.

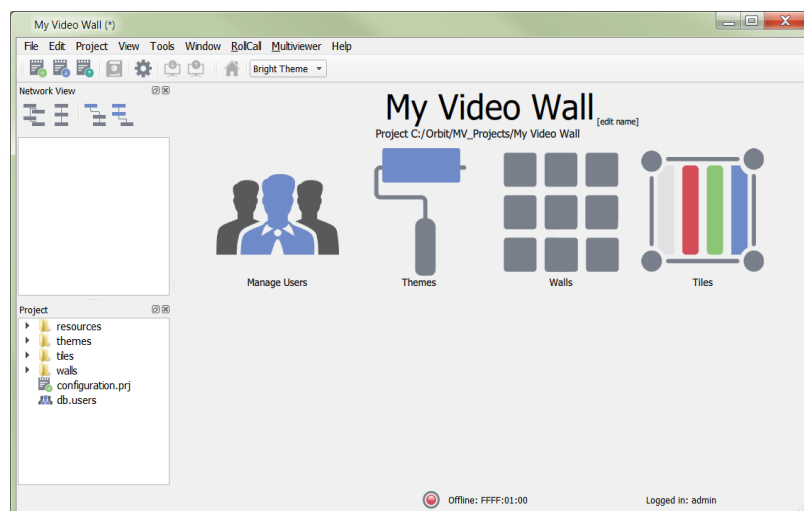


Figure 220 Orbit Project Screen

Set up the project users:

4. Click on the **Manage Users** icon.
5. Change the password of the default user 'admin'.
6. Add two new users. For example, new user 'Supervisor' with password 'supervise' and new user 'Operator' with password 'operate'.

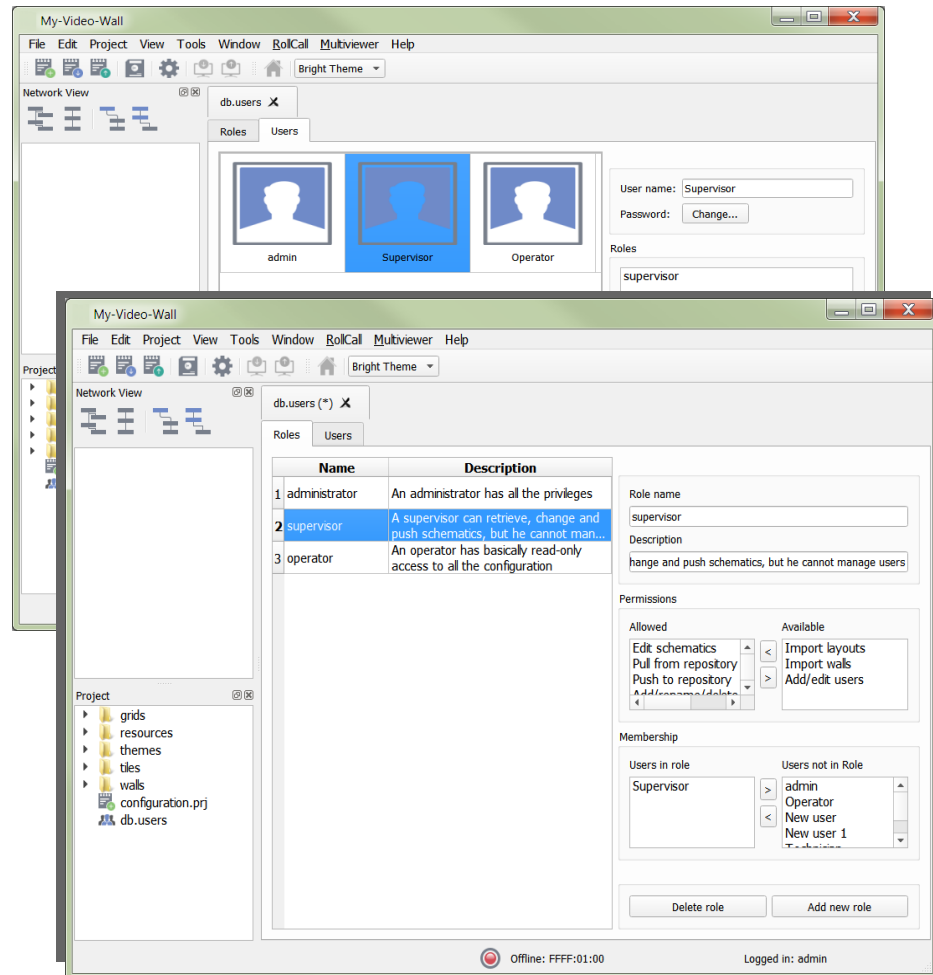




Figure 221 Orbit Manage Users Tab - Roles and Users

7. Click the **Save File** icon, , to save changes and close the tab, .

Note:

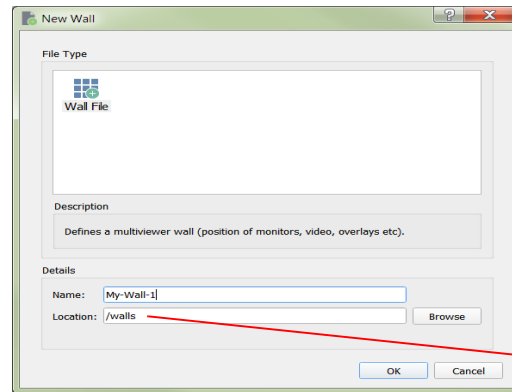
The management of users of a project enables project settings to be controlled and protected when the project is deployed. Different users are given different levels of access to the project, i.e. different roles.

User names and roles can be changed by an administrator.

8.7.2 Create the Video Wall

New Wall:

1. Click the **Wall** icon in the Orbit Project screen.
Click **New Wall**.



Location within the project folder

Figure 222 Orbit New Wall Dialog

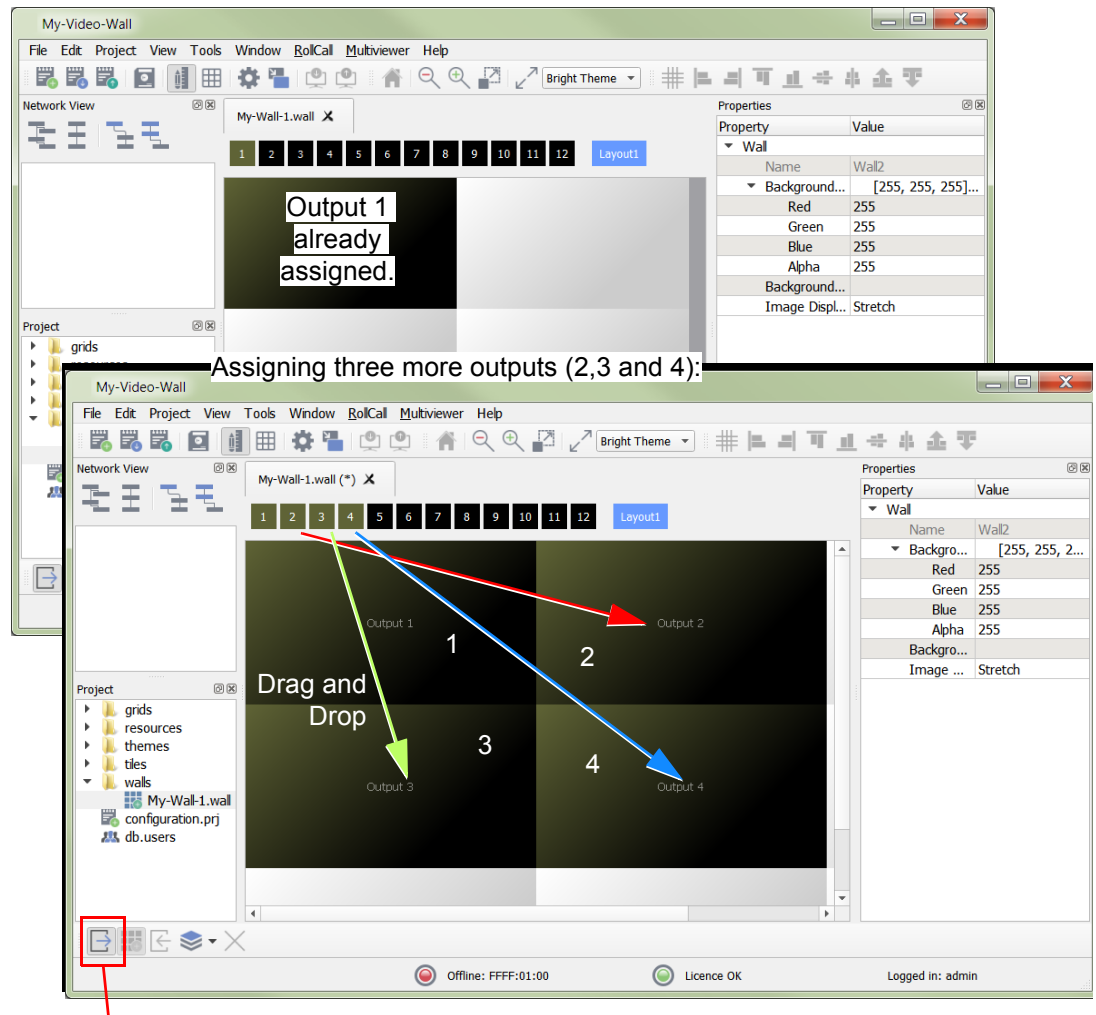
2. Ensure Orbit is operating in 'Design Mode'.
(Clicking the **Design/Run-Mode** icon toggles between Orbit's 'Design Mode' and 'Run Mode'.)



Figure 223 Orbit Design/Run-Mode Icon's Appearance

Assign Outputs to a Wall:

3. Select the **Assign Outputs** icon and Drag and Drop the first four multiviewer outputs (1,2,3 & 4) onto the wall in a 2x2 arrangement, see Figure 224.



Assign Outputs icon selected


Figure 224 Orbit Wall - Assigning Outputs

Note:

Note the **Properties Box** on the right-hand side of the screen.

The wall property **Name** is set to "Wall1".

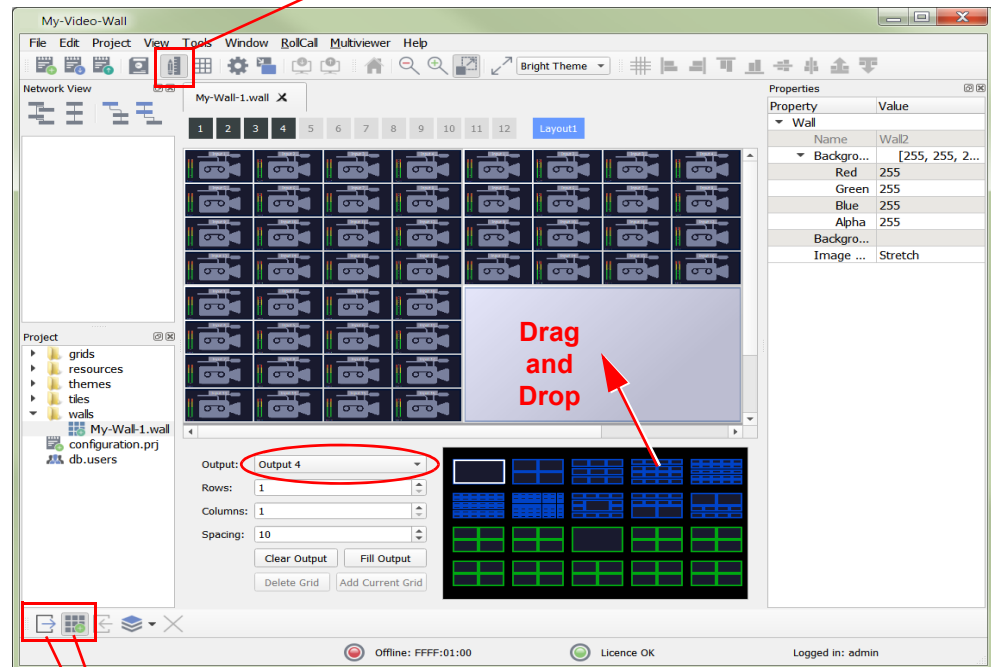
This property value is used in the RollCall **Layout** screen, see Section 2.12.4.1 "Multiviewer RollCall Layout Screen Overview" on page 71.

4. Click the **Save File** icon, , and deselect the **Assign Outputs to Wall** icon.

Add a Tile Grid to Wall:


5. Select **Add Tile Grid to Wall** mode, see Figure 225.
Select each output in turn in the **Output** drop-down box and drag a 4x4 tile grid onto each output screen.

Design mode selected



'Assign Tile Grid to Wall' icon selected
and 'Assign Outputs' icon deselected

Figure 225 Orbit Wall - Added 4x4 Tile Grids

6. Click the **Save File** icon, , to save changes.
7. Select tiles and drag the corners to increase their size, see Figure 226.

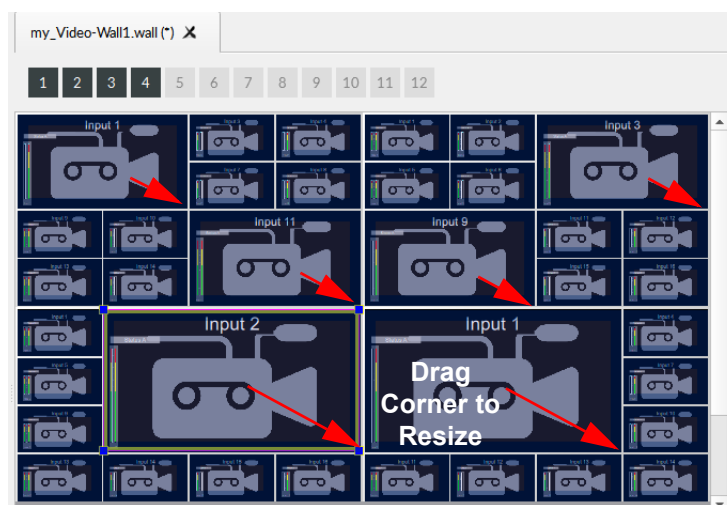


Figure 226 Orbit Wall - Drag Tile Corners to Resize Tiles on the Tile Grid

Change some Tiles on the Wall:

8. Click on the top-left tile to select it.
In the **Properties Pane**, change **Type** from 'Video Tile' to 'Analogue Clock'.
9. Repeat previous step for the top-right tile, but changing **Type** to 'Digital Clock'.
Two clocks are now on screen, see Figure 227.

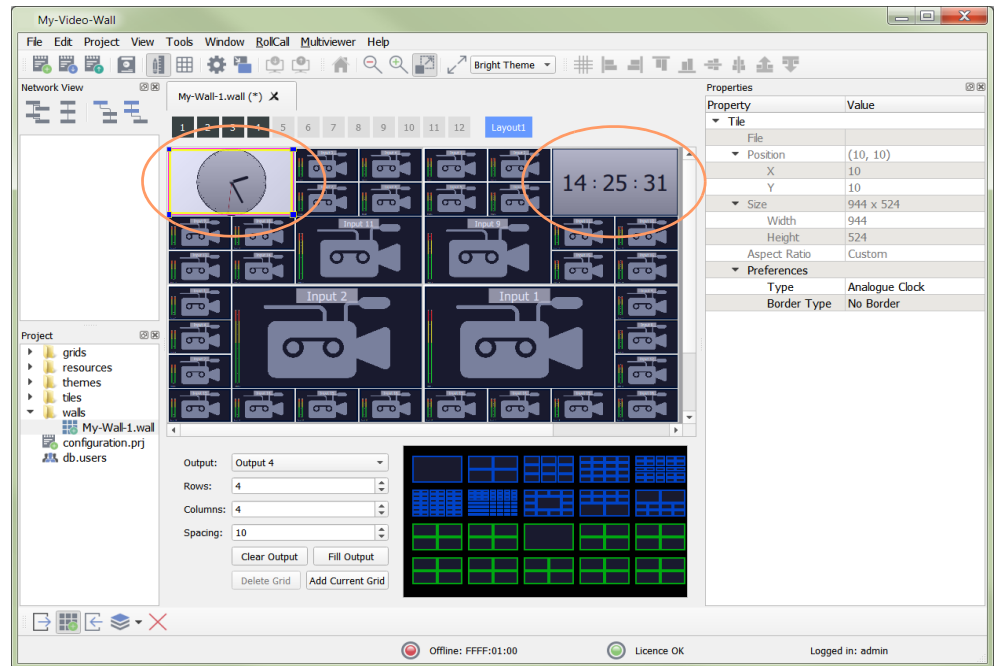


Figure 227 Orbit Wall - Resized Tiles on the Tile Grid

10. Select the right-hand largest size video tile and, in the **Properties Pane**:
 - Deselect **Left Audio Bar**.
 - Select **Right Audio Bar**.
 - Set **Right Audio Bar Number** to 8.
 - Set **Audio Bar Position** to 'Outside Video'.
 - Select **UMD**.

Repeat these steps for the large tile above, to yield two changed tiles. See Figure 228. The format of the video tiles has been changed, which can be seen by comparing the left- and right-hand side larger video tiles in Figure 228.

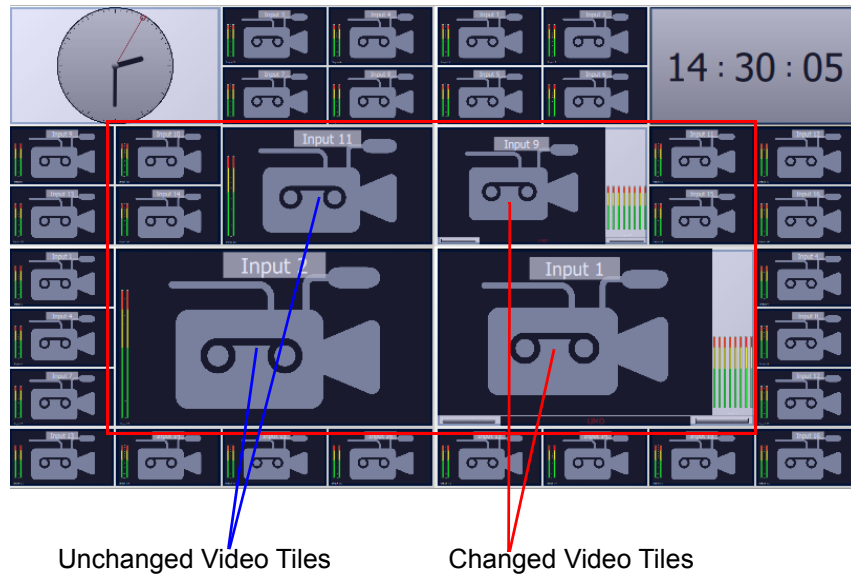



Figure 228 Orbit Wall - Different Video Tile Formats

11. Click the **Save File** icon, , to save changes.
12. Leave the wall tab open.

Note:

Auto-Tiles:

The video tiles used in this sub-section are the default tiles which are automatically generated by Orbit for multiviewer applications. They are called **Auto-Tiles**.

An **Auto-Tile** comprises video, audio bars and UMD widgets.

The styles of each of these widgets can be changed when the tile is selected in the **Wall Editor**. Thus individual **Auto-Tiles** can be tailored to show different styles. (Styles are defined as part of the theme applied to the Orbit project.)

8.7.3 Import a Logo Image into the Project

A .png or .jpg image file may be used in a project:

1. In the **Project View**, expand the 'Resources' sub-folder to reveal the 'Images' sub-folder.
2. To import a graphics image file into the 'images' sub-folder, do either:
 - Drag and drop the image file from Windows Explorer into the Orbit **Project View** 'images' sub-folder.
 or
 - Right-click on the 'images' sub-folder in the **Project View** and click **Import File(s)**.
Browse to a graphics image file, select the file. and click **Open**.

Note: If the image file will overwrite an existing resource file, the user is prompted. Click 'Yes' to import and overwrite.

The graphics file is imported into the Orbit project and placed into the 'images' sub-folder. It is shown in the **Project View** under the 'Resources' folder, see Figure 229.

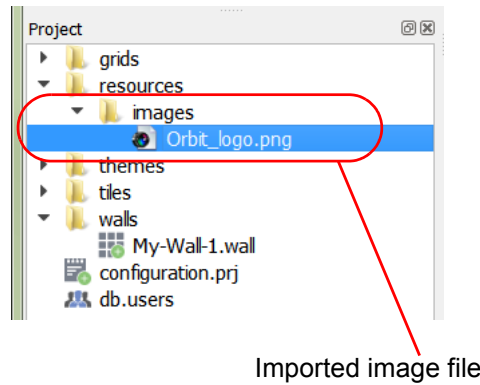


Figure 229 Orbit Imported Graphics Logo file

The imported graphics file is now available for use in the Orbit project.

Note:

If the resource is already used in the project, close down any tiles or walls and then re-open them in order to use the newly-imported resource.

8.7.4 Create Custom Tiles

8.7.4.1 Create a Simple Custom Tile

1. Either:

- Click on the **Tiles** icon in the Orbit Project screen. And Click **New Tile**.

Or

- Right-click on 'tiles' in the **Project View**. And Click **New Tile**.

Then select 'Empty Tile' and name the tile.

2. In the **Tile Editor** screen, click in the tile area and:

- Set property Width to 800 and Height to 100.
- Set the background to gray (Extended Style > Background > Background Color).

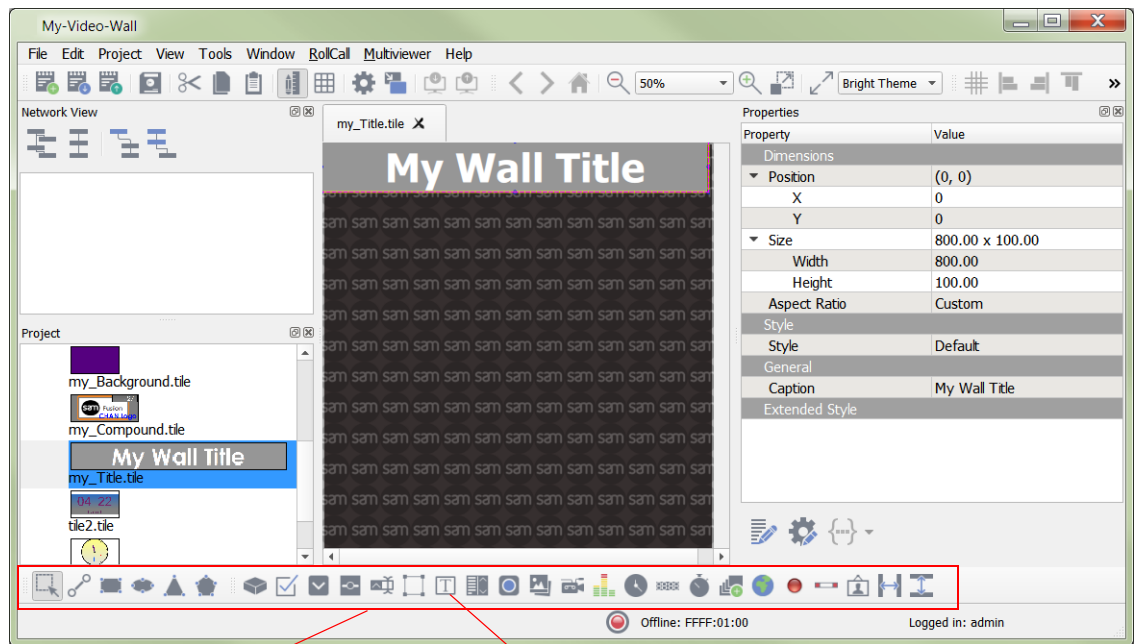
This sets the tile size and sets its color to gray.

3. Add a **Label** from the **Widget Tool Bar** along the bottom of the screen. Select the added label and modify its properties:

- Set the **Caption** text required (General > Caption).
- Resize the label to fill the tile area. (Drag a corner of a selected widget to resize.)
- Set Extended Style property **Horizontal Alignment** to 'Center'.

See Figure 230.


4. Click **Save File**.



Widget Tool Bar

Label Widget Icon

Figure 230 Orbit Custom Tile - 'my_Title.tile'

5. Close the 'tile' tab, .

This has created a simple custom tile.

8.7.4.2 Create a Compound Custom Tile

1. Create a new tile.
2. In the Tile Editor screen:
 - Set property **Width** to 500 and **Height** to 200.
 - Set property **Background Color** to gray.
3. Add a Label, an Image and a Digital Clock widget from the **Widget Tool Bar** along the bottom of the screen.

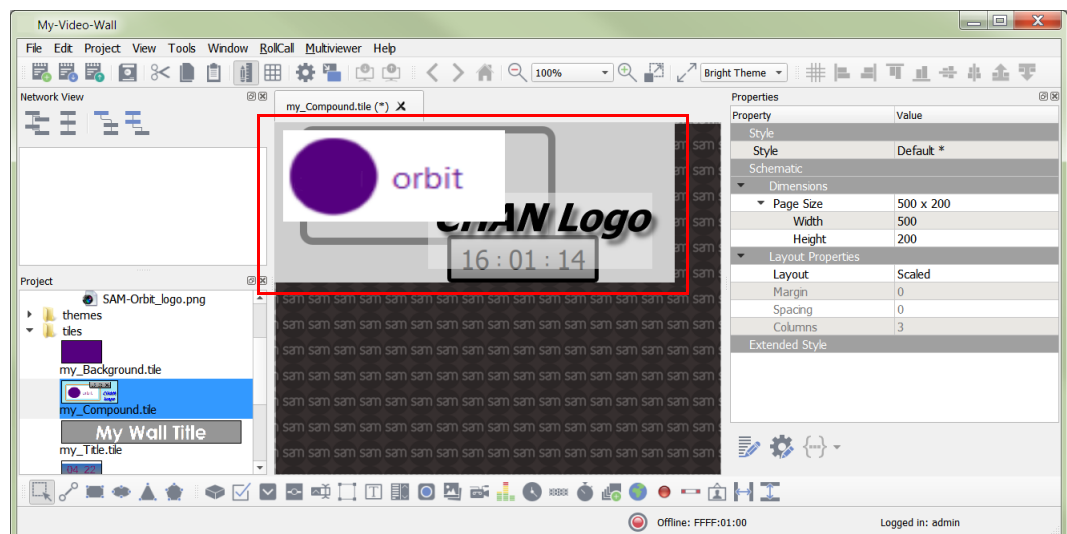


Figure 231 Orbit Custom Tile - 'my_Compound.tile'

4. Select the **Image** widget in the new tile.
In the **Properties Box**, set extended style image property **Source Path** to be the path to the logo file imported into the Orbit project's 'Resources' folder.
5. Select the **Label** widget in the new tile.
In the **Properties Box**, set extended style font property **Color** to be Blue.
6. Select the **Digital Clock** widget in the new tile.
In the **Properties Box**, deselect the extended style general property **Caption Visible** (Extended Style > General >Caption Visible).
7. Adjust the size, position and layering of these widgets on the tile.
With a widget selected, right-click to set the **Ordering** (to front, to back, forward, backward). See Figure 232.

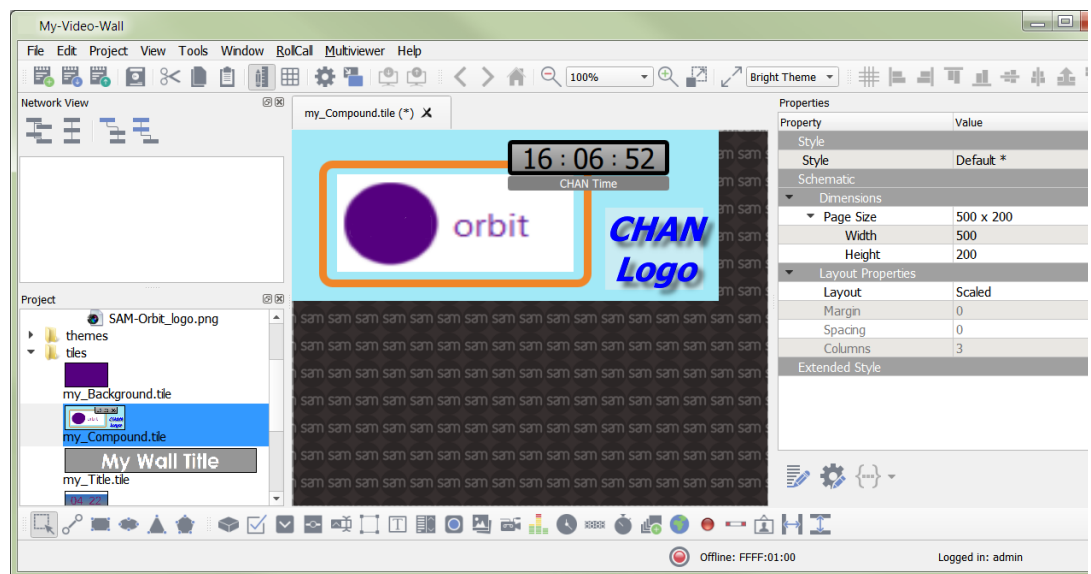




Figure 232 Orbit Custom Tile - completed 'my_Compound.tile'

8. Click the **Save File** icon, , to save changes.
9. Close the 'tile' tab, .

This has created a custom tile comprising three widgets arranged and layered on a tile.

8.7.5 Add Custom Tiles to the Wall

1. Return to the wall tab.
(To reopen this tab: Go to the **Project View** and expand the 'walls' item.)
2. Double-click on the wall name item (for example, 'My_Video-Wall1.wall').
The wall appears in the **Wall Editor** screen.
3. In the **Project View**, expand the 'tiles' item.
Drag and drop the two new custom tiles onto the wall. See Figure 233.

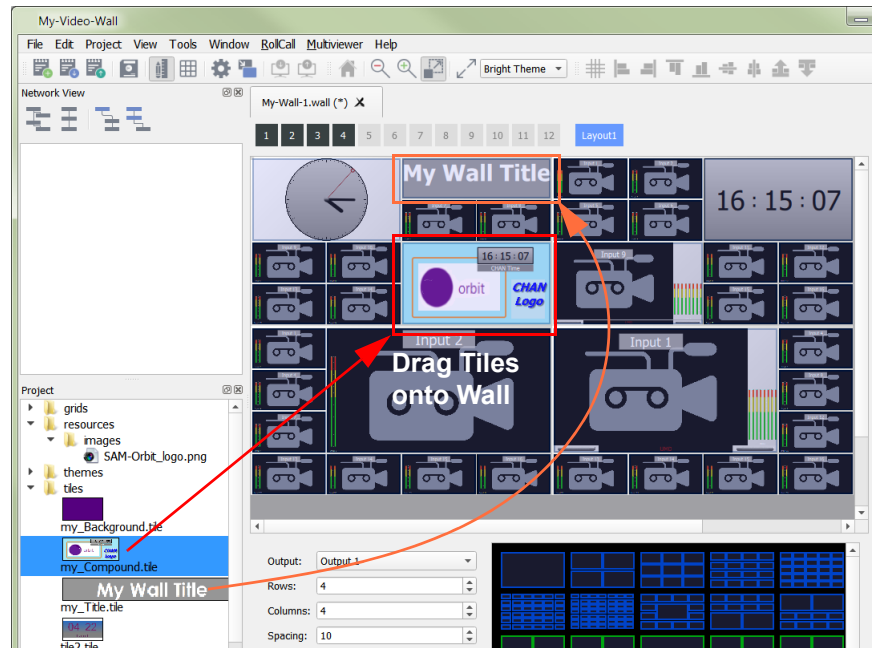


Figure 233 Custom Tiles added to Wall

Note:

Tile Icons - Each custom tile is listed in the **Project View** pane, in the 'tiles' sub-folder. Each tile is represented by a tile icon, generated when a tile is saved.

Tile sizes and positions may be adjusted in **Fine Grid** mode:

- Click the **Add Tile Grid to Wall** icon (bottom left of Orbit screen) to toggle between **Tile Grid** and **Fine Grid** Wall Editor modes. When the icon is deselected, the **Wall Editor** is in **Fine Grid** mode, see Figure 234.

Add Tile Grid to Wall icon is *deselected*

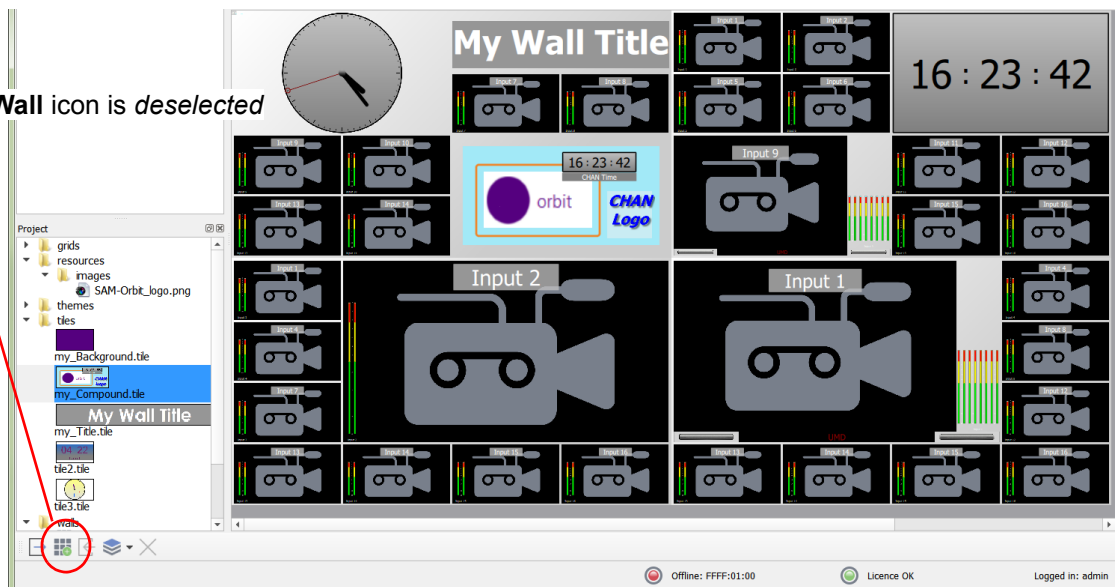


Figure 234 Fine Grid Wall Editor

To fine-adjust a wall tile's size and position in **Fine Grid** mode:

- Zoom in on the **Wall** with the zoom controls in the main tool bar.
- Select the required **Tile** and drag the **Tile** and/or its corners as required.

See Figure 235 for an example of "before" and "after" fine adjustment of the **Tile**.

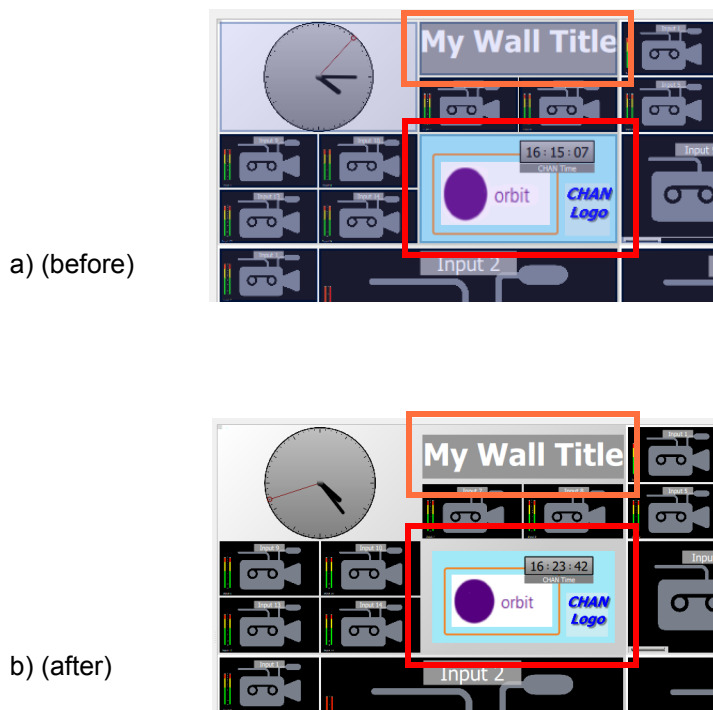


Figure 235 Fine Adjustment of Custom Tile Size and Position
 a) Original unadjusted tile (Before)
 b) Fine-Adjusted tile size and position (After)

7. Click the **Save File** icon, , to save changes.

A custom tile has now been added to the wall.

8.7.6 Assign Multiviewer Inputs to Wall Video Tiles

Having laid out a basic video wall, multiviewer video inputs (Input 1, Input 2 etc) need to be assigned to each video tile:

1. Click outside the wall in the **Wall Editor** area. See Figure 236. This deselects anything already selected.
2. Select all the video tiles, omitting the clock tiles. Refer to the '**Selection Help**' note, below, to select all the video tiles, as shown in Figure 236.

Note:

Wall Editor Selection Help:

| | |
|----------------------------------|--|
| Deselect All | Click outside the wall area in the Wall Editor. |
| Select | Click on an item to select it. |
| Add Select | Shift-click on an item to add it to the current selection. |
| Remove from Selection | Shift-click on an already-selected item to remove it from the current selection. |
| Select All | 'Ctrl-A' selects all items. |
| Select all within an area | Shift click and drag to define a rectangular area. |

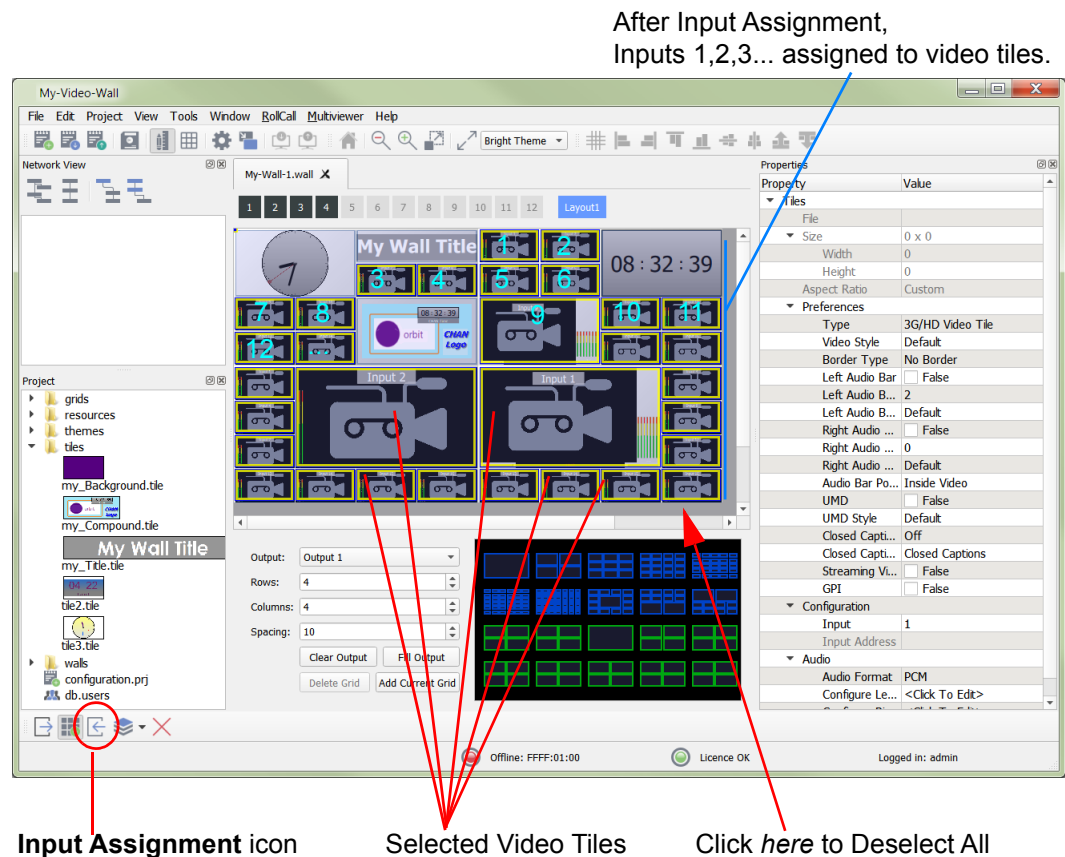


Figure 236 Multiviewer Inputs Assigned to Video Tiles

- Click the **Input Assignment** icon.
In the Input Assignment dialog, set 'Input Starting From:' to 1.
Click **OK**.
Video input numbers are assigned to each video tile, from 1 onwards.

The **Input Assignment** step may take 2 to 3 seconds to complete. While this is in progress, a "Loading" splash message is displayed, see Figure 237.



Figure 237 "Loading" Splash Message.

Note: When the **Input Assignment** step is complete, the caption text (e.g. "Input 1") in the video tiles may not fully reflect the input assignment. This is normal.

The caption text is actively updated on the video wall at run-time, when the video wall is deployed on a multiviewer.

- Click the **Save File** icon, , to save changes.

8.8 Push Project to the Multiviewer

8.8.1 Before Pushing a Project

Before a project can be pushed to a multiviewer, the project must be set up with some details about the target multiviewer.

If the project has been created as a new 'connected project', or it has been pulled from a multiviewer, then the multiviewer details will be set up in the project. These will require changing to target a different multiviewer.

If the project is new, then multiviewer details must be set up.

See Section 1.3 "Setting up Multiviewer Device Details in a Project" on page 4.

8.8.2 Check Project Name

Use a unique project name for each multiviewer. Change the name if the project is being pushed to a different multiviewer.

8.8.3 Push the Project to the Multiviewer

To push a project back to the Multiviewer:

1. Click **Project > Select Multiviewer**.
Enter the **IP address** of the Multiviewer.
2. Click **Project > Push**.
The modified project is pushed to the multiviewer.

The multiviewer adopts the new wall layout.

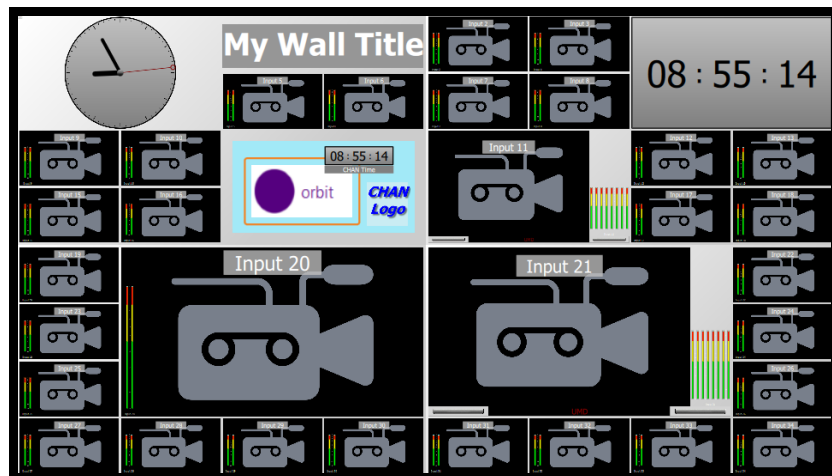


Figure 238 Basic Video Wall

8.9 Wall Layouts

Different video wall layouts may be stored in one multiviewer Orbit project. Switching between wall layouts enables a video wall installation to be easily used for different purposes.

1. Open an existing project and double-click on a wall design in the **Project View**. The project's wall design appears in the **Wall Editor**. See Figure 239.

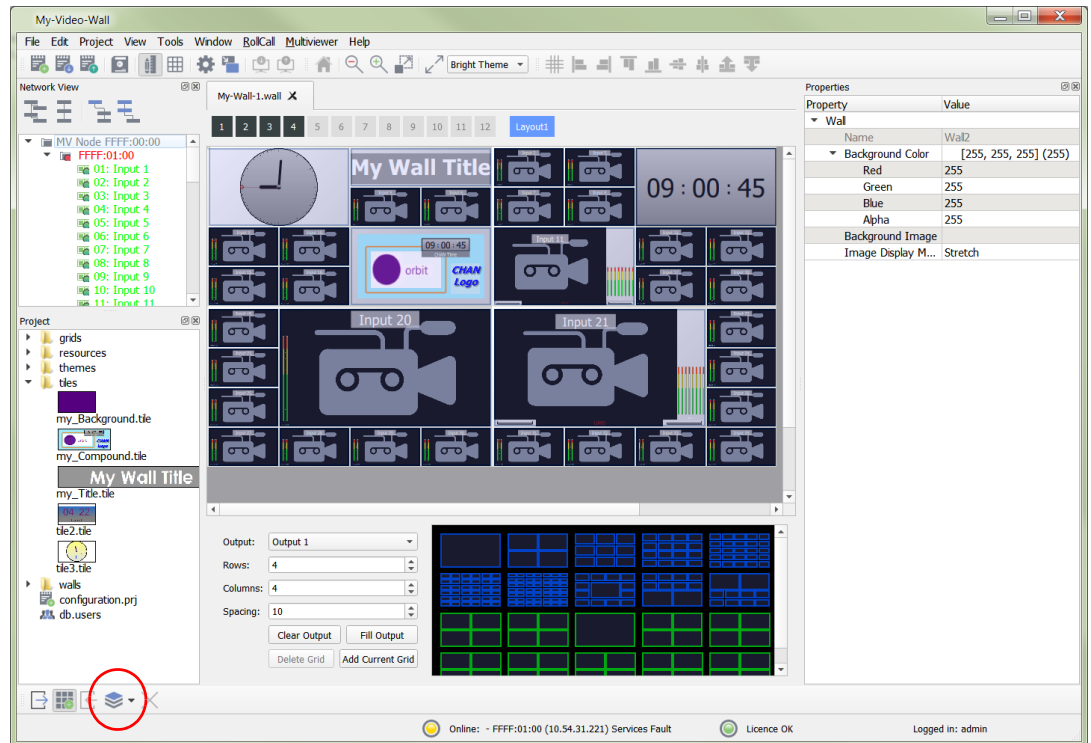



Figure 239 New Layout icon in Wall Editor Screen

2. Click on the **Layout** icon in the **Wall Editor**, bottom left. Click on **New Layout**

This creates a blank wall layout for the current wall design, with the same multiviewer outputs assigned.

3. Add a Tile Grids and Tiles etc to the wall to create your new second layout. See Figure 240 for an example of a second wall layout.

4. Click the **Save File** icon, , to save changes.

5. Click the **Layout** icon and select the layout required to change between wall layouts.

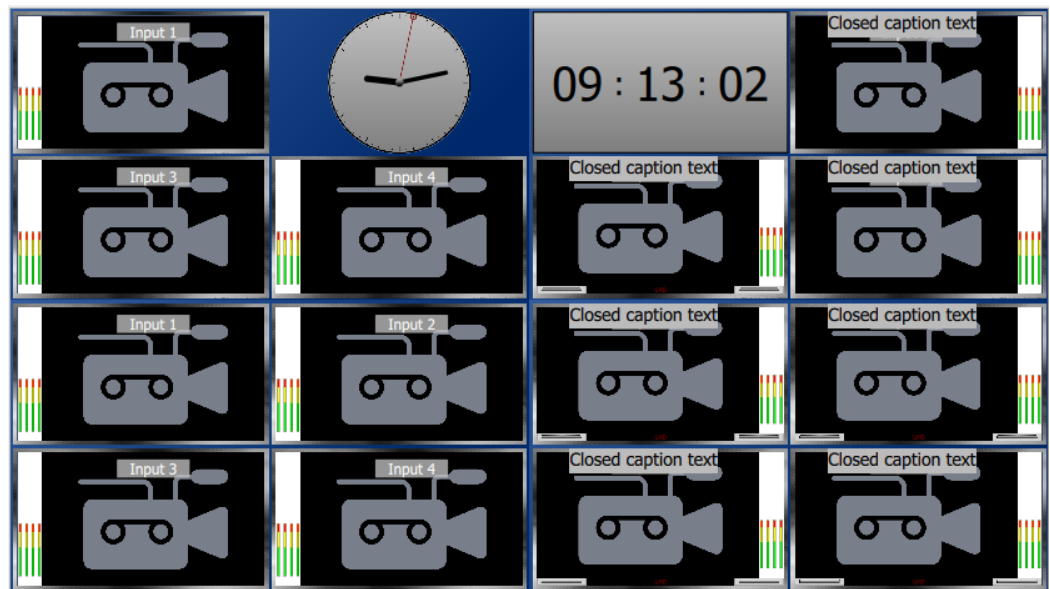


Figure 240 Example Second Wall Layout, "Layout 2"

Note:

The wall layouts are called "Layout1", "Layout2" etc. These are used in the RollCall Control Panel's **Layout** screen, see Section 2.12.4.1 "Multiviewer RollCall Layout Screen Overview" on page 71.

8.10 Themes

When placed onto a wall, a widget's appearance can be tailored to suit user-preferences by assigning a widget **Style** to it. Versions of a widget can be created with specific properties, creating widget **Styles which** are alternate versions of a widget (differing in style and appearance only; functionally, the widget is unchanged).

A **Theme** is a set of widget styles which may be applied to a wall.

Using **Themes**, a multiviewer monitor wall layout can be made to suit a studio house style. This is done in a multiviewer project with 'themes' in the Orbit application.

8.10.1 Baseline Default Theme

A new Orbit project has a factory-set theme, which is listed in the **Project View** as "**Default.theme**" in the 'themes' sub-folder.

This **Default theme** is a baseline theme. A user can edit this theme, if required, or create a new theme. It is recommended that, if changes are required to the widget styles used, then create a new theme in the **Themes Editor** and then make the widget style changes required.

8.10.2 Creating Themes

Users may create their own themes:

1. In the **Project View**, right-click on the 'themes' item. Click **Add Theme**. Enter a name for the new theme. Click **OK**.
2. Click **Yes** to the question about using the new theme as the current theme. The **Themes Editor** screen is shown.

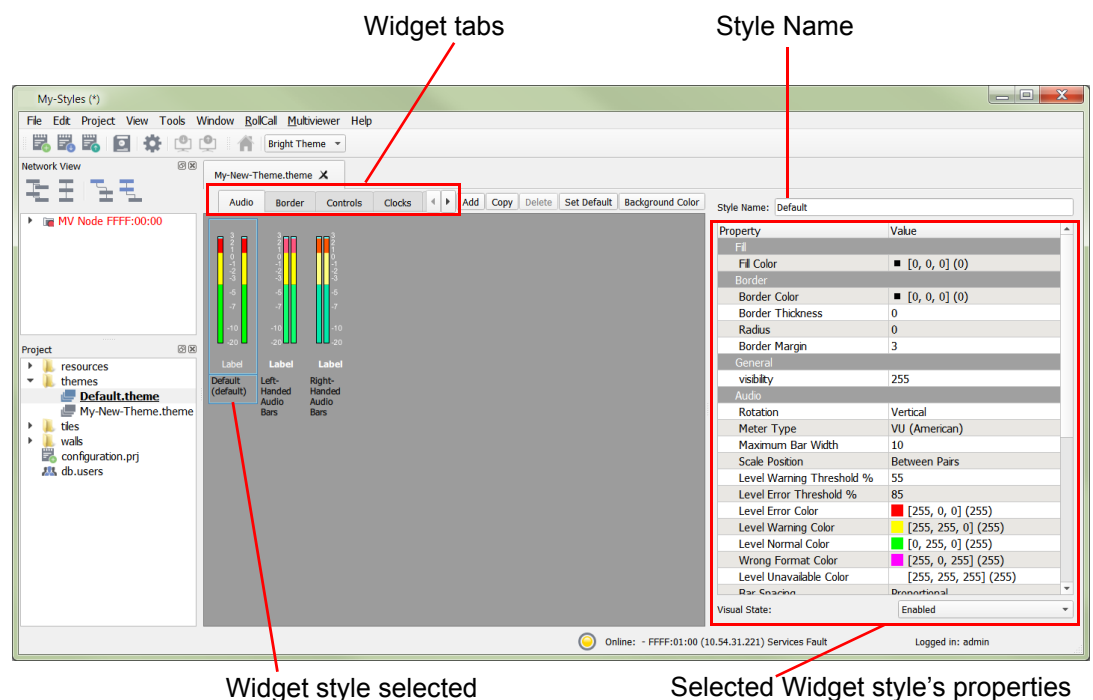



Figure 241 Orbit Theme Editor Screen

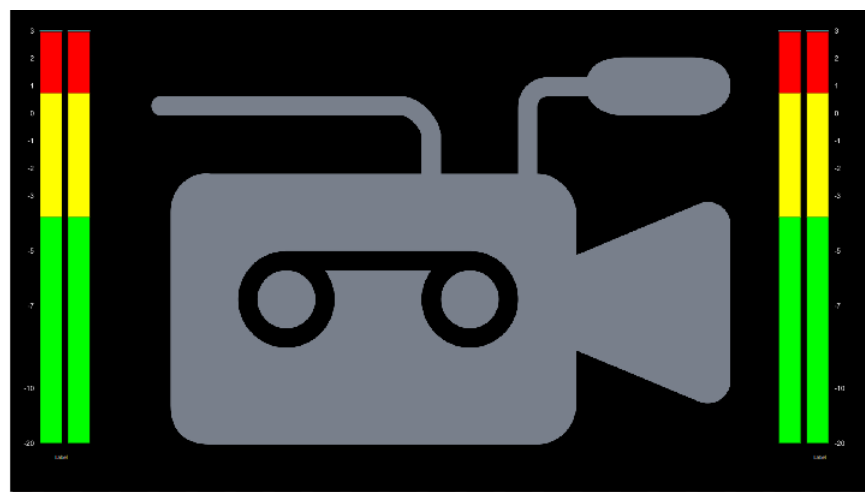
In the **Themes Editor**, each different widget type has a tab along the top of the screen.

3. Select a widget type tab to display the different widget styles for that widget type

4. Click on a widget style to select it.
The widget style's properties are shown in the **Properties Pane**.
5. Click **Add** at the top of the editor screen to create a new widget style.
A new version of the widget appears.
6. Enter a name for the new widget version in the **Style Name** box.
7. The widget properties for this widget version (new style) may now be changed as required.
8. Click the **Save File** icon, , to save changes.

A widget can have more than one style.

For example, this then allows the user to define a “left-handed” and a “right-handed” Audio Bars widget for use on a tile, see Figure 242.




Left-handed style of Audio Bars

Right-handed style of Audio Bars

Figure 242 Video Tile with Audio Bars and different Current Project Themes applied.

The widget a style that is to be used when a particular theme is applied must be defined. The association between the widget style and a theme is indicated in the **Themes Editor** by the suffix “(default)” after the widget style name.

Note: The suffix “(default)” indicates the default widget style to use for a theme. It is not related to the ‘Default theme’.

9. Select a widget style in the Themes Editor
and click **Set Default** to set that widget style to be the default style for that particular theme.
The label “(default)” is appended to the widget style name.
This shows that it is the default widget style to use for the theme being edited.
Repeat this for other widgets.
10. Click the **Save File** icon, , to save changes.

More than one theme may be created but only one theme may be applied to a project at a time. All widget styles may need to be changed to suit a particular theme.

Any theme may be applied to a project but only one theme at a time.

8.10.3 Applying Themes to a Wall

Themes are applied to an Orbit project in the Orbit application. When a theme is applied to a wall, it is used as the current theme for that wall.

To apply a theme:

1. In the **Project View**, expand the 'themes' item.
Right-click on a theme name (.theme) to select it.
2. Click **Set as Current Theme**.

A project's wall layout can have themes applied to produce a different look-and-feel to one or a widget type, see Figure 243, or to all widgets in an entire wall, see Figure 244.

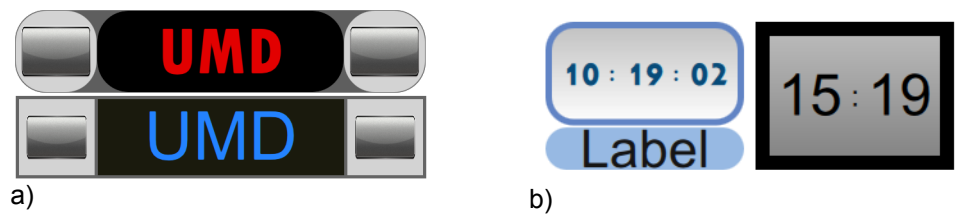


Figure 243 Two Widget Styles for: a) UMD; b) Digital Clock

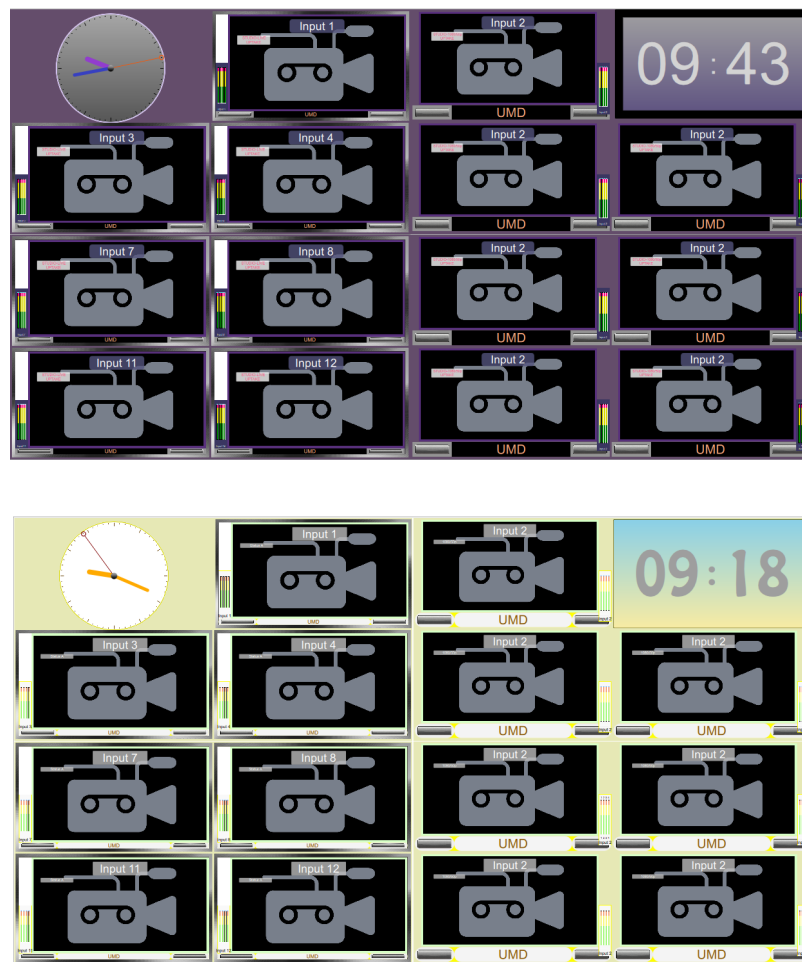


Figure 244 Two Themes Applied to an Entire Wall

8.10.4 Applying a Style to a Widget

If a specific style is required for a widget on a wall, then it can be set up in the widget properties for each widget on the wall.

Custom Tile:

When widgets are dragged onto a **Custom Tile**, the style they use may be selected in the properties box when the widget is selected in the **Tile Editor**.

Auto-Tile:

The style used by widgets on an Auto-Tile can be selected in the properties box when the Auto-Tile is selected in the **Wall Editor**.

8.10.5 Locally Modifying the Style of a Widget

After a widget is placed on a tile and its style property has been set, its style can be modified in situ. Modifications are then only applied to the particular widget on the tile, and not to other widgets. The widget's style property value is then shown with "*" appended, to indicate that the widget style has been modified locally. For example 'Clock_House-Style *', see Figure 245.

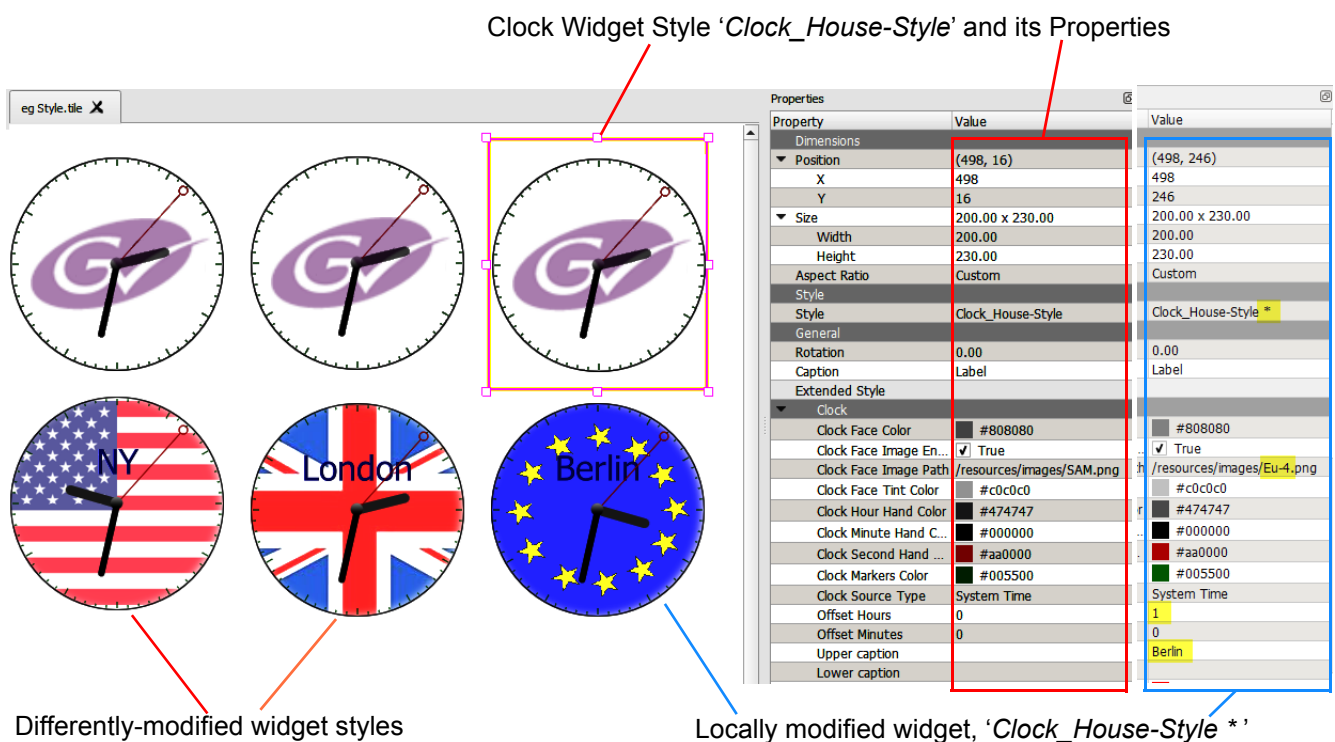


Figure 245 Locally Modified Widget Style

If a locally-modified widget style needs to be kept for re-use, it can be saved back into the main widget style, thus updating all widgets using that style, or it can be saved as a new widget style.

To save a locally modified style as a new style:

1. Right-click on the locally-modified widget and select **'Styles > New Style'** in the drop down menu.
2. Enter a name for the new style.
3. Click **OK**.

To update a style that has been locally modified:

1. Right-click on the locally-modified widget and select '**Styles > Update Style**' in the drop down menu.
2. Click **OK**.

8.10.6 Default Style

For each widget type in a theme, a default style is designated.

When a theme is applied to a project, the theme's styles get applied to all the widgets on each wall of the project. If a specific style has been set up for a widget, then the style is looked for in the theme being used. If this is not present in the current theme, then the theme's designated **default** style for the widget type is used.

9 Examples - UMDs and Border Tally

Examples - UMDs and Border Tally

| | |
|--|-----|
| Using a Built-in UMD Widget | 230 |
| Set Up The Behaviour | 231 |
| Set Up the Binding | 233 |
| Do-It-Yourself UMD (Lamps and a Label) | 236 |
| Configuring Behaviours and Bindings | 237 |
| Lamp Widgets | 241 |
| Label Widget - Border Color | 246 |
| Label Widget - Font Color | 249 |
| Label Widget - Caption Text | 251 |
| D-I-Y UMD Behaviour and Bindings Recap | 254 |
| Another Do-It-Yourself UMD (Shape Fill and Images) | 255 |
| Configuring Behaviours and Bindings | 256 |
| Shape Widget | 258 |
| Image Widget | 261 |
| Label Widget | 262 |
| D-I-Y UMD Behaviour and Bindings Recap | 263 |
| Border Tally Example | 264 |
| Configuring Behaviours and Bindings | 265 |
| Border Behaviour and Binding | 270 |
| Label Widget | 273 |
| Border Tally Behaviour and Bindings Recap | 278 |
| Alternative Border Tally | 280 |
| Exercise the Tile and Widgets | 283 |

This section contains some example uses of Orbit's **Behaviours** and **Bindings** for making UMDs or displaying UMD information with a border tally.

Orbit offers a UMD widget and this is typically used to show TSL protocol information on-screen.

A UMD widget is built into multiviewer video **Auto-Tiles** and the functionality is already set up. However, for **Custom Tiles**, a UMD widget needs to be added and the required functionality formed with **Behaviours** and **Bindings**.

Alternatively, a customized UMD or some other form of displaying TSL protocol information may be required. In this case, other widget types can be used and the required functionality must be formed with **Behaviours** and **Bindings**.

Note:**Selection/ Deselection of graphical items when editing items in the Tile Editor:**

| Select / Deselect | Action |
|------------------------------|---|
| Deselect All | Click outside the schematic area. Or press the ' Esc ' keyboard key. |
| Select | Click on an item to select it. Alt-Click selects another item: <ul style="list-style-type: none"> • In a stack of items. • Or selects the Group Layout that an item belongs to. |
| Add Select | Shift-Click on an item to add it to current selection. Shift-Alt-Click to add another item from a stack of items to the current selection. |
| Remove from Selection | Shift-Click on an <i>already-selected</i> item to remove it from the current selection. |
| Select All | 'Ctrl-A' selects all items. |
| Select an Area | Shift drag cursor on screen to describe a rectangular area. All objects wholly or partly in the area will be selected. |

9.1 Using a Built-in UMD Widget

This example adds a UMD widget to a **Custom Tile** and adds the required **Behaviours** and **Bindings**: In this case, there is:

- A **Behaviour** which specifically reads in the external TSL information and provides the required TSL data internally (**TSL Behaviour**).
- A **Binding** which specifically applies TSL data to a UMD widget (**TSL Binding**).
- A UMD widget.

Figure 246 depicts the Behaviour-Binding-Widget connectivity required.

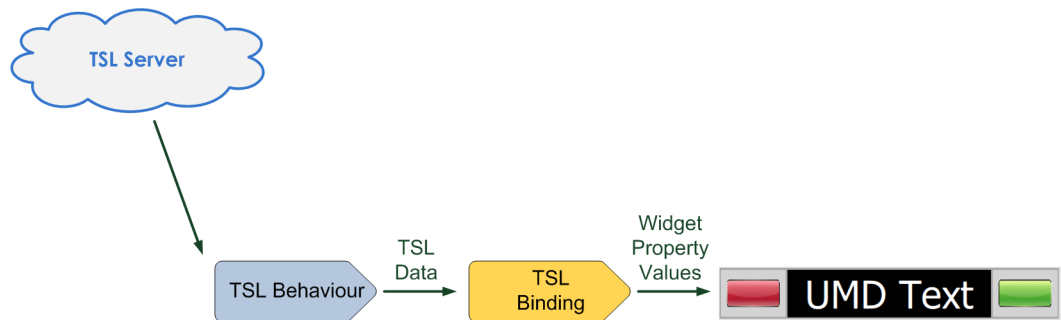


Figure 246 Connectivity for a UMD Widget

To construct the **Behaviours** and **Bindings** for a UMD widget:

1. Create a new empty tile, for example, a quarter-sized tile, and add a UMD widget.

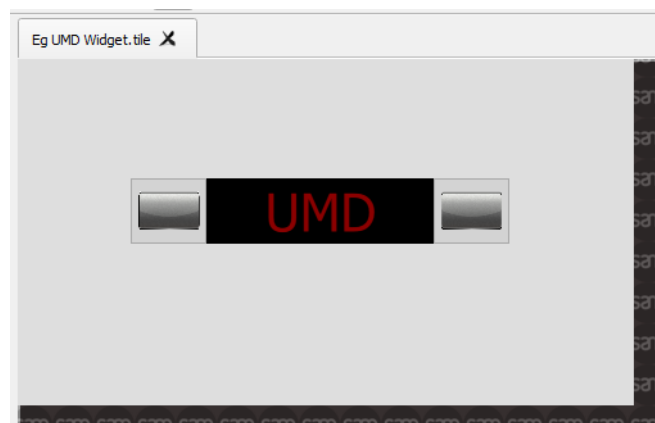


Figure 247 New Tile with UMD Widget

2. Click the **Save** icon in the **Main Menu** to save any changes to the **Tile**.
3. Open the graphical **Behaviours and Bindings Editor** for the **Tile** by clicking the **Edit Behaviours** icon in the Tile Editor's Advanced Tool Bar.



4. Select the UMD widget.
The **Behaviours** and **Bindings** screen for the selected widget is shown. There are no **Behaviours** nor **Bindings** set up yet.

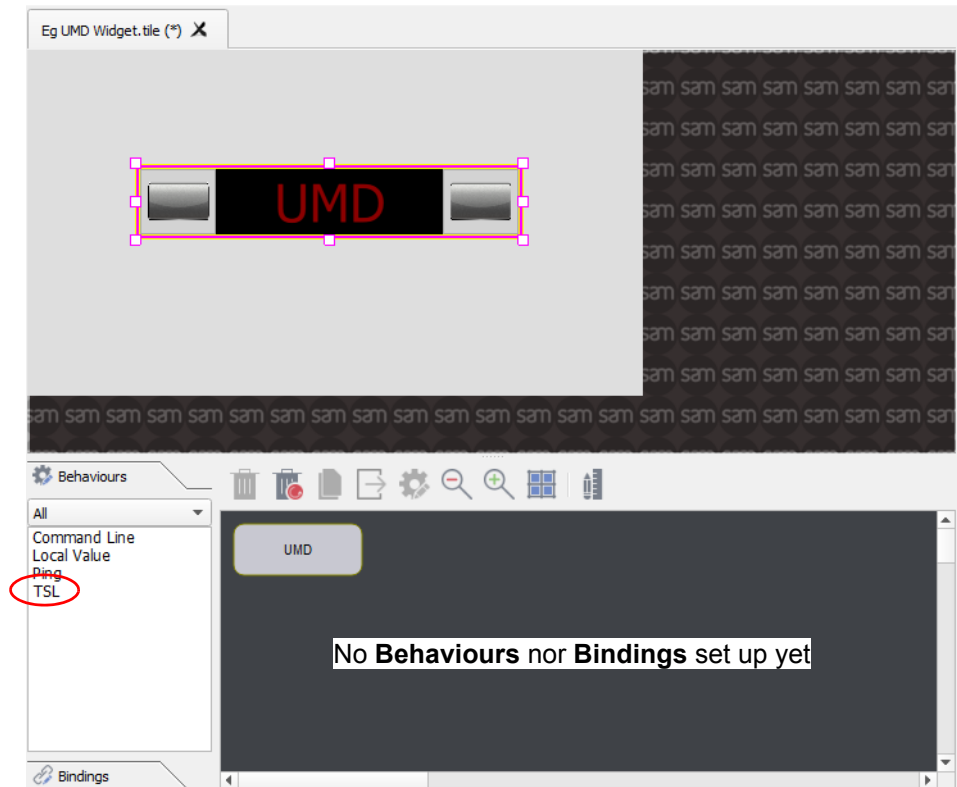


Figure 248 Behaviour and Bindings Editor, UMD Widget Selected

5. Click on the **TSL** name in the list of **Behaviours**, to add a **TSL Behaviour** to the editor screen. A **TSL UMD Binding** block is also automatically added.

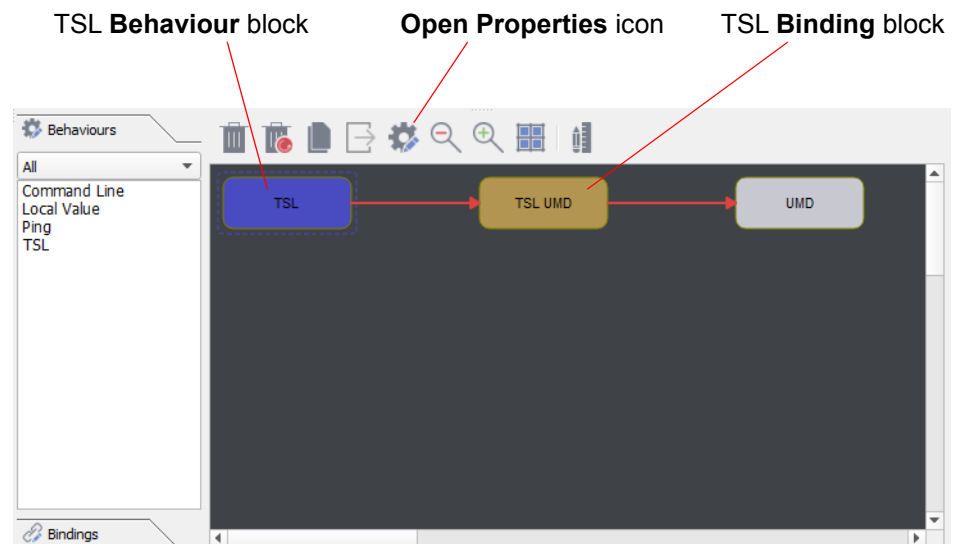


Figure 249 TSL Behaviour and Binding

9.1.1 Set Up The Behaviour

6. Select the **TSL Behaviour block** and then:
 - Click on the **Open Properties icon** , see Figure 249.

Or

- Double-click on the **TSL Behaviour** block.

The **Properties** dialog for the **Behaviour** is shown. See Figure 250.

| Property | Value |
|----------------------|-------------|
| Name | My TSL 5.0 |
| RollCall Address | {Address} |
| TSL Version | Version 5.0 |
| 5.0: Screen Index | 0 |
| 5.0: Display Index | 0 |
| 3.1: Display Address | 0 |

Figure 250 TSL Behaviour Properties Dialog

7. Change property **Name** to 'My TSL 5.0'.
8. Change property **TSL Version** to 'Version 5.0'.

The two TSL version 5.0 **Binding** properties '5.0: Screen Index' and '5.0: Display Index' may now be edited. Each is currently set to a fixed number, '0'.

In order that the property value can be changed each time the tile is used, a Orbit system variable may be created and used for each property value:

9. Select property '5.0 Screen Index'.
Click the **Select Variable** 'down-arrow' icon, see Figure 251.
And select 'Create Variable from Property'.

| Property | Value |
|----------------------|-------------|
| Name | My TSL 5.0 |
| RollCall Address | {Address} |
| TSL Version | Version 5.0 |
| 5.0: Screen Index | 0 |
| 5.0: Display Index | 0 |
| 3.1: Display Address | 0 |

Figure 251 'Creating a System Variable'

10. The 'Enter Variable Name' dialog is shown. See Figure 252.

Figure 252 Enter Variable Name Dialog

11. Click **OK**.

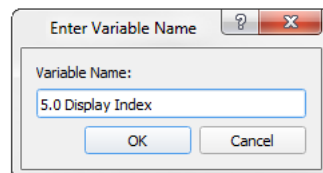
Note:

In this example, the variable name is left as the default variable name, as shown in the initial **Enter Variable Name** dialog.

Leaving a variable name as the default name means that the TSL property can be assigned at wall-level by Orbit with the **Input Assignment** icon.

If the variable name is set to some other value, then manual assignment of the TSL property is required.

12. Select property '5.0: Display Index'.
Click the **Select Variable** 'down-arrow' icon.
And select 'Create Variable from Property'.
The 'Enter Variable Name' dialog is shown. See Figure 253.

**Figure 253 Enter Variable Name Dialog**

13. Click **OK**.

The **TSL Behaviour** properties have been set up, see Figure 254.


| Edit My TSL 5.0 | |
|---|-------------------|
| Property | Value |
| Name | My TSL 5.0 |
| RollCall Address | {Address} |
| TSL Version | Version 5.0 |
| 5.0: Screen Index | 5.0 Screen Index |
| 5.0: Display Index | 5.0 Display Index |
| 3.1: Display Address | 0 |
| <div style="text-align: right;">  Close </div> | |

Figure 254 Completed TSL Behaviour Properties Dialog

14. Click **Close**.

9.1.2 Set Up the Binding

Proceed to address the **TSL UMD Binding** properties:

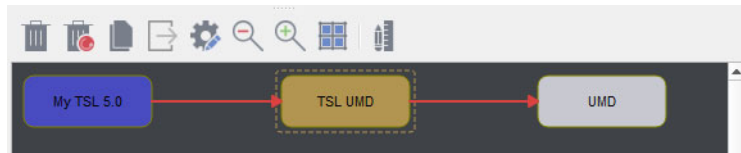


Figure 255 'My TSL 5.0' Behaviour

15. Select the **TSL UMD Binding** block and click the **Open Properties** icon. Or double-click the **TSL UMD Binding** block.

The **Properties** dialog for the **Binding** is shown. See Figure 256.

Figure 256 TSL UMD Binding Properties Dialog

16. Change the **Name** to 'My TSL UMD'.
17. The **Source Behaviour** property is automatically set to be the 'My TSL 5.0' **Behaviour** in this case.

Note: If some other **Behaviour** is to be the source for this **Binding**, then click **Select** and choose from the **Behaviours** listed.

18. Click **Close**.

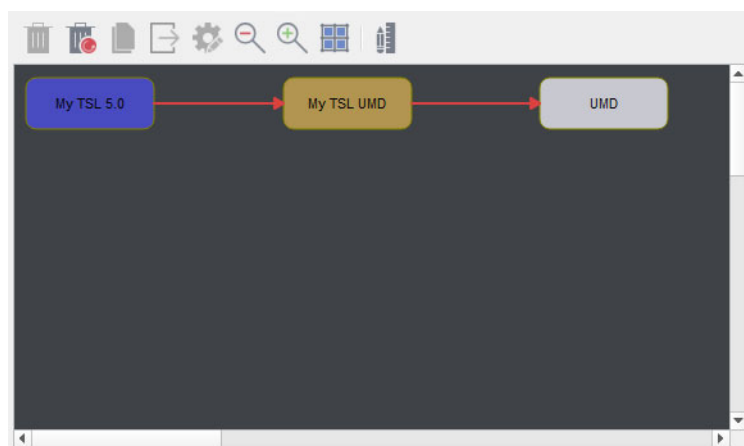


Figure 257 'My TSL 5.0' Behaviour and 'My TSL UMD' Binding

19. Click the **Save** icon in the Main Tool Bar. See Figure 258.

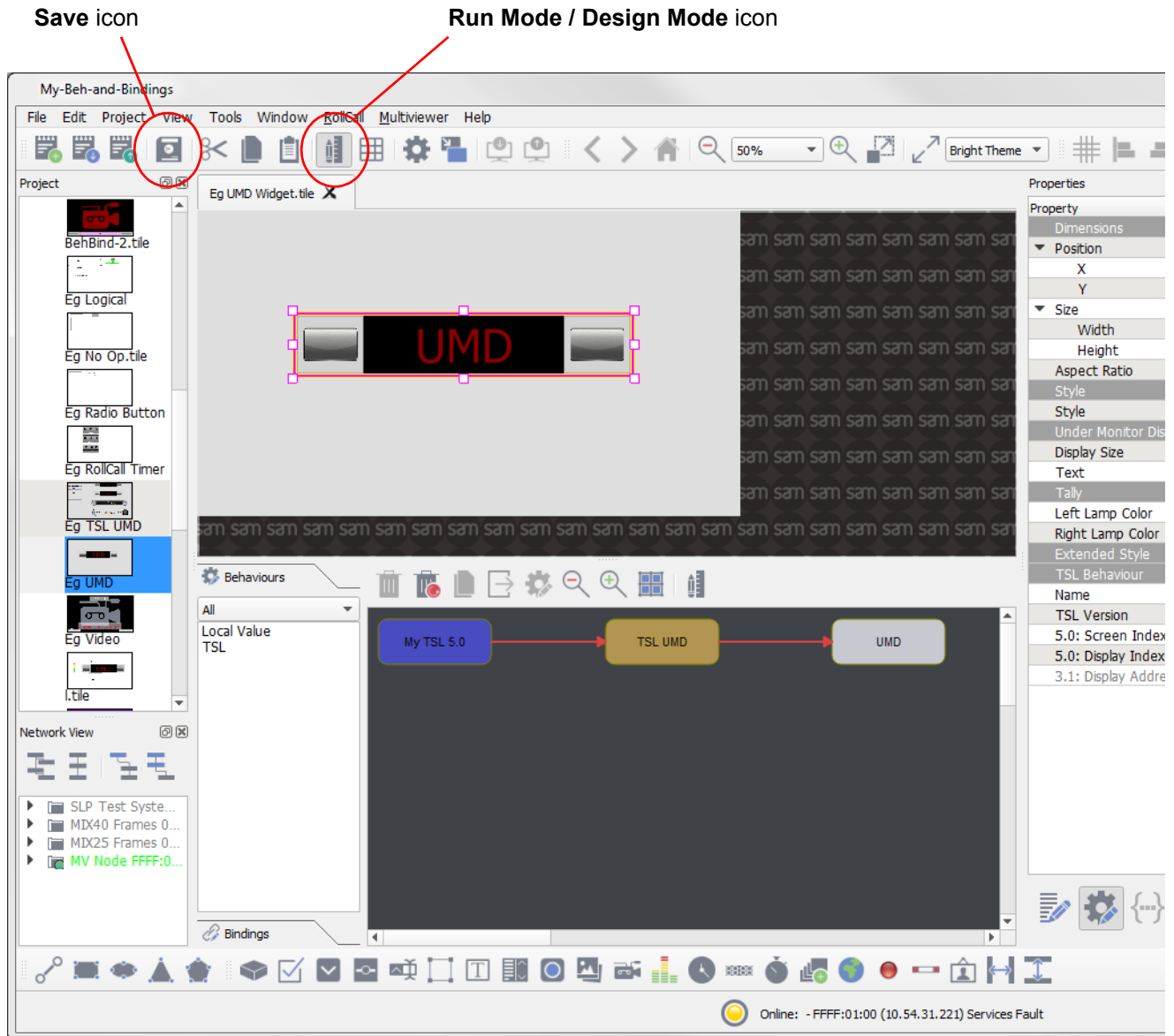


Figure 258 'My TSL 5.0' Behaviour and 'My TSL UMD' Binding

Configuration for the UMD widget is done.

The **TSL Behaviour** (a TSL Listener) gets data from a TSL server in the system. The **TSL Behaviour** presents information about a certain video feed to the **TSL UMD Binding**, which connects the data to the UMD widget in a predetermined way.

9.2 Do-It-Yourself UMD (Lamps and a Label)

In some cases, a customized UMD will be required. This can be built from separate, individual widgets and configured to have the required functionality - a “Do-It-Yourself UMD” (D-I-Y UMD).

This example uses:

- A **TSL Behaviour**.
- A **Property Binding**.
- Lamp and Label widgets.

Figure 259 compares the UMD widget to an example “Do-It-Yourself” UMD made from widgets.

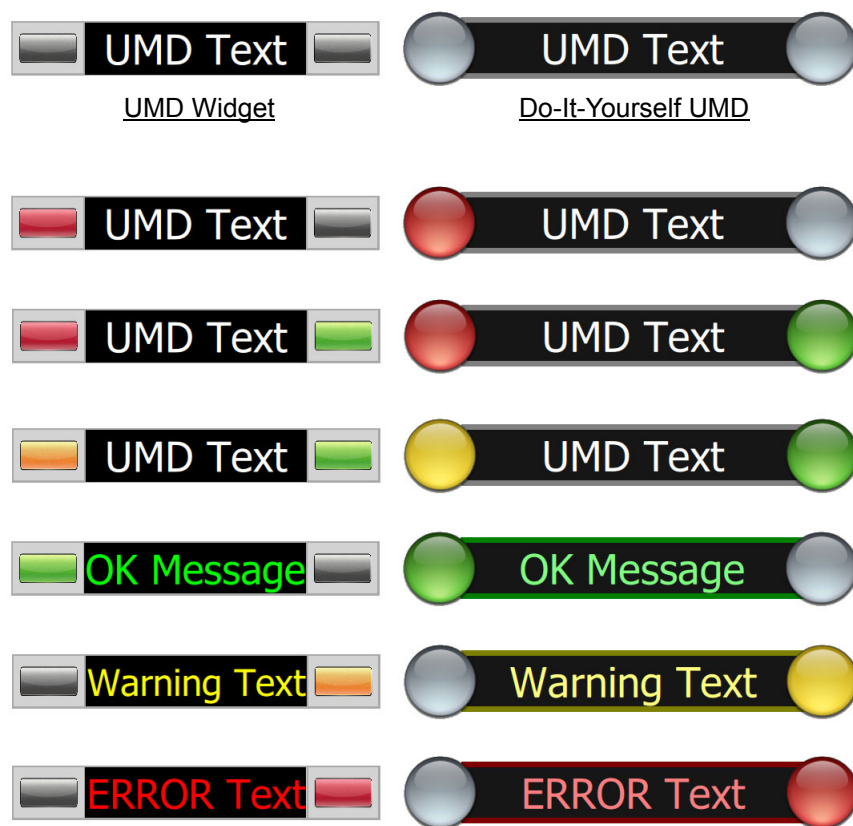


Figure 259 ‘Comparing UMD Widget and Do-It-Yourself UMD Operation

The example **D-I-Y UMD** comprises two Tally Lamp widgets and a Label widget, as shown in Figure 260.

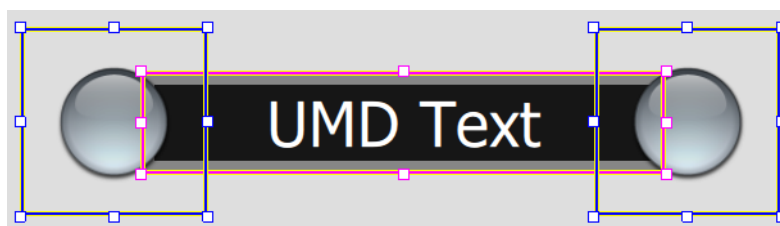


Figure 260 ‘Widgets of a Do-It-Yourself UMD

TSL Protocol Version 5.0 will be used in this example and each component widget will be set up to react to some TSL system data:

- **Left-hand Tally Lamp** - To change color according to the TSL 'Left Lamp' data value.
- **Right-hand Tally Lamp** - To change color according to the TSL 'Right Lamp' data value.
- **Label** widget - This is required to: Show the TSL Label text data, and to change font color and border edge color according to the TSL 'Text Tally' data value.

The required **Behaviour** and **Binding** connectivity is shown in Figure 261.

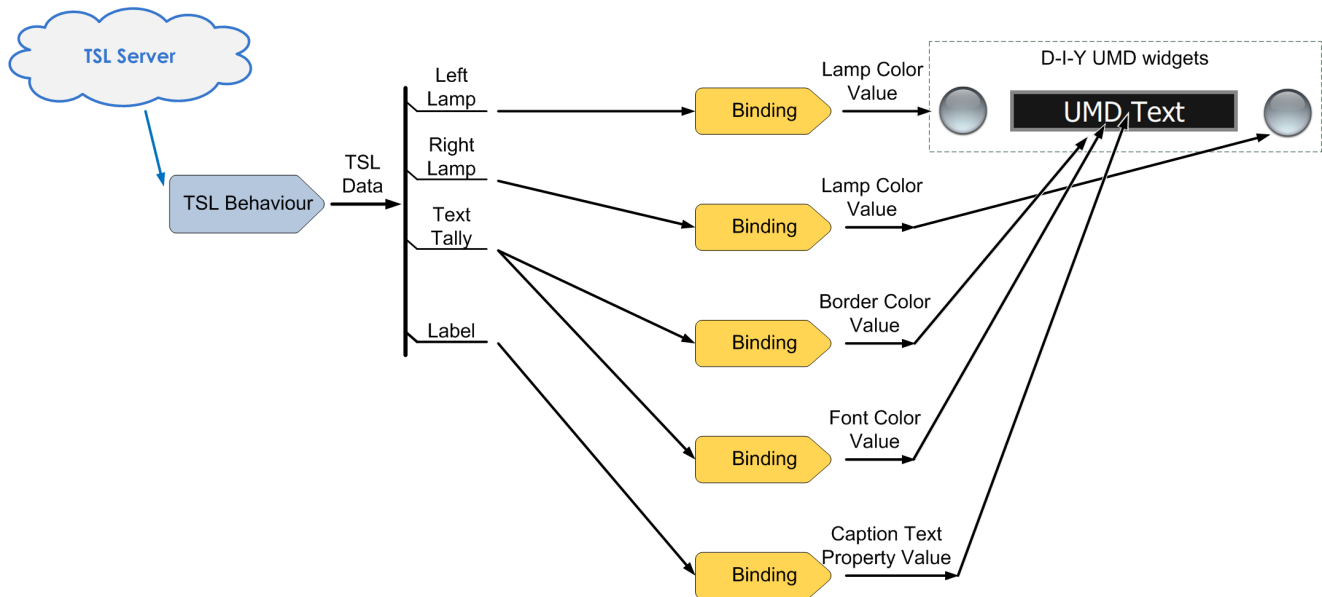


Figure 261 Connectivity for Do-It-Yourself UMD

The TSL Behaviour's TSL data is used several times by the widgets and is common to them.

9.2.1 Configuring Behaviours and Bindings

A description of how to configure the functionality of this **D-I-Y UMD** using **Behaviours** and **Bindings** follows:

1. Create a new empty tile, for example, a quarter-sized tile, and form a UMD from two Tally Lamp widgets and a Label widget. See Figure 260.
2. Open the graphical **Behaviours and Bindings Editor** by clicking the **Edit Behaviours** icon in the Tile Editor's Advanced Tool Bar.



3. Click on the **Tile** to *deselect* any widgets. This has, by default, selected the **Tile** itself. The **Behaviours** and **Bindings** screen for the **Tile** itself is shown. There are no **Behaviours** nor **Bindings** set up yet. See Figure 262.

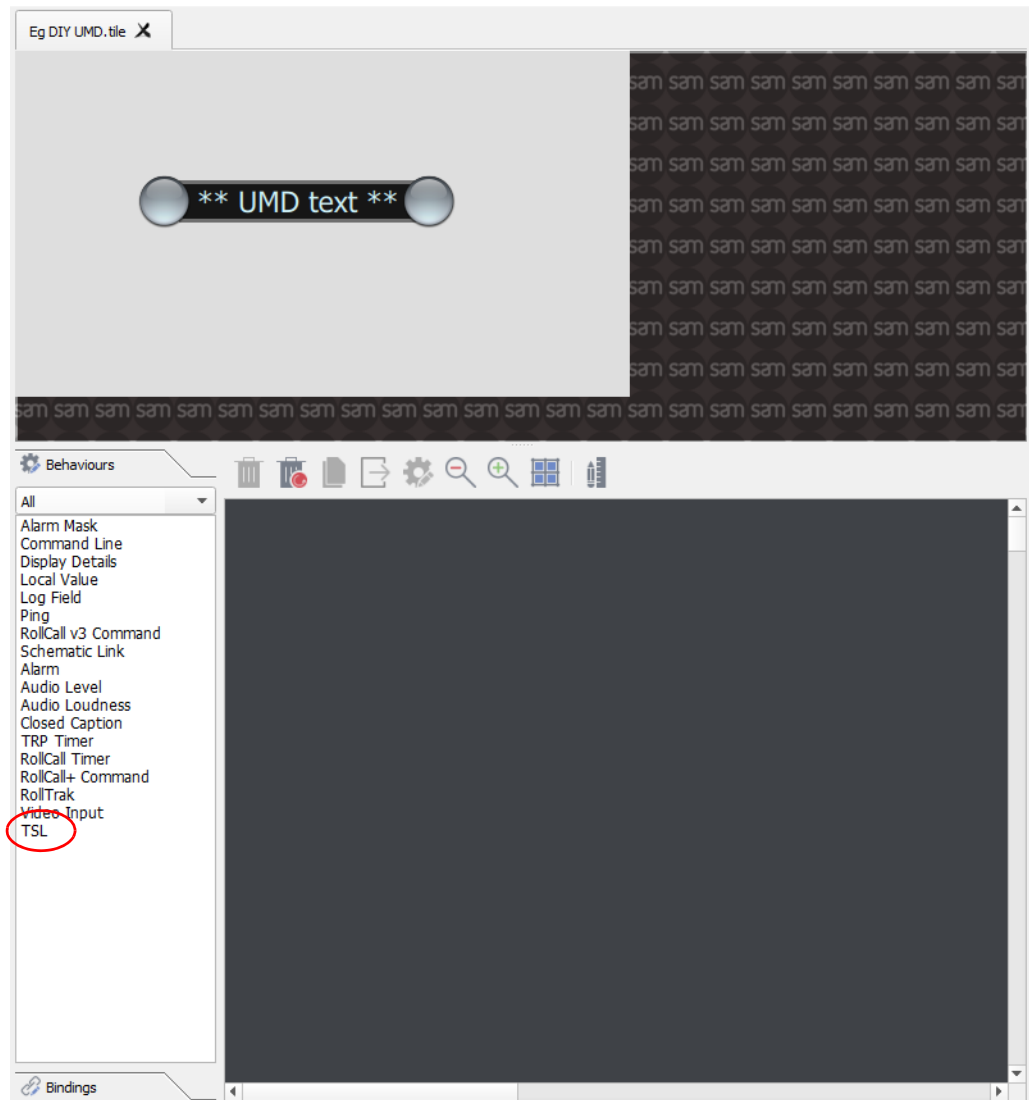


Figure 262 Behaviour and Bindings Editor

Initially, we add a **Behaviour** and a **Binding** to the **Tile**, at 'tile-level', before moving on to each component widget:

4. Click on the **TSL** name in the list of **Behaviours**.
A **TSL Behaviour** block is added to the editor screen.

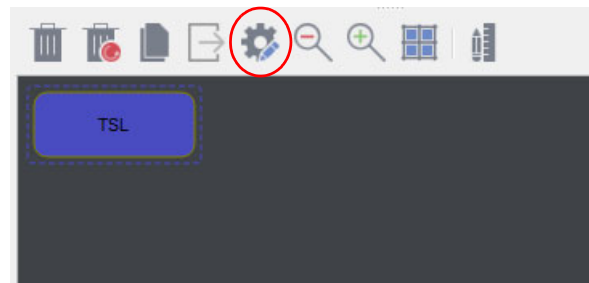


Figure 263 TSL Behaviour Block

5. Select the **TSL Behaviour** block and click on the **Open Properties** icon, see Figure 263.
Or double-click on the **TSL Behaviour** block.
The **Properties** dialog for the **Behaviour** is shown.
6. Change property **Name** to 'My TSL 5.0'.

7. Change property **TSL Version** to 'Version 5.0'.

The other two TSL version 5.0 **Binding** properties, '5.0 Screen Index' and '5.0 Display Index', may now be edited. Each is initially set to a fixed number, '0'.

In order that the property values can be changed each time the tile is used, a variable may be created. The variable may be used within the **Tile/Schematic** on which it is defined, i.e. it has tile/schematic-level scope.

To create variables for each property value:

8. Select property '5.0 Screen Index'.
Click the **Select Variable** 'down-arrow' icon, see Figure 264.
Select 'Create Variable from Property'.

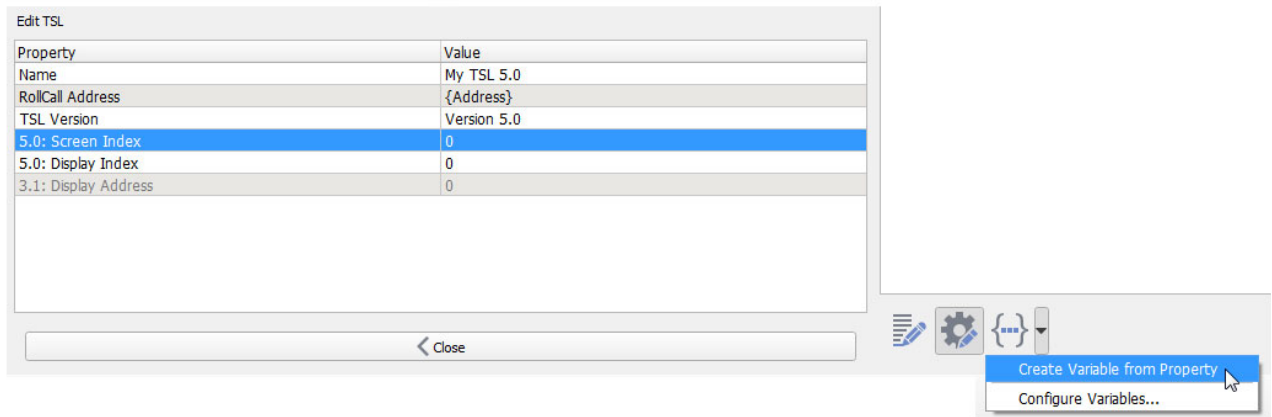


Figure 264 'Creating a System Variable

9. The 'Enter Variable Name' dialog is shown. See Figure 252.

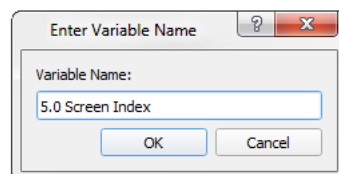


Figure 265 Enter Variable Name Dialog

10. Click **OK**.

Note:

In this example, the variable name is left as the default variable name, as shown in the initial **Enter Variable Name** dialog.

Leaving a variable name as the default name means that the TSL property can be assigned at wall-level by Orbit with the **Input Assignment** icon.

If the variable name is set to some other value, then manual assignment of the TSL property is required.

11. Select property '5.0: Display Index'.
Click the **Select Variable** 'down-arrow' icon.
And select 'Create Variable from Property'.
The 'Enter Variable Name' dialog is shown. See Figure 253.

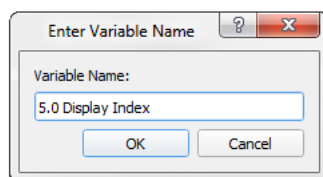


Figure 266 Enter Variable Name Dialog

12. Click **OK**.

The **TSL Behaviour** properties have been set up, see Figure 267 for the completed properties dialog.

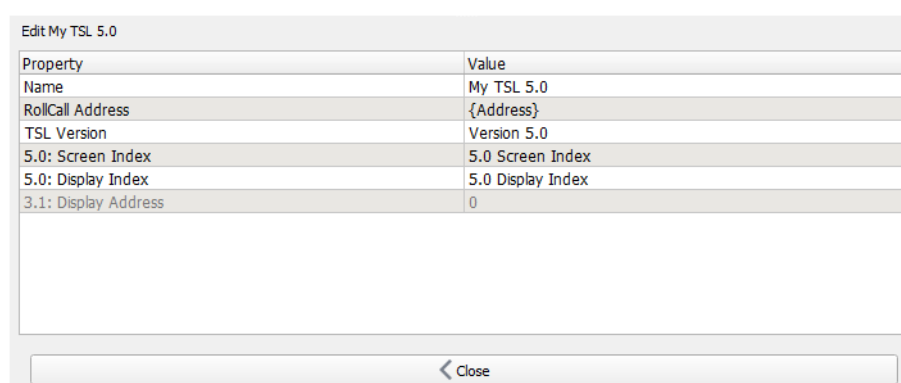


Figure 267 Completed TSL Behaviour Properties Dialog

13. Click **Close**.

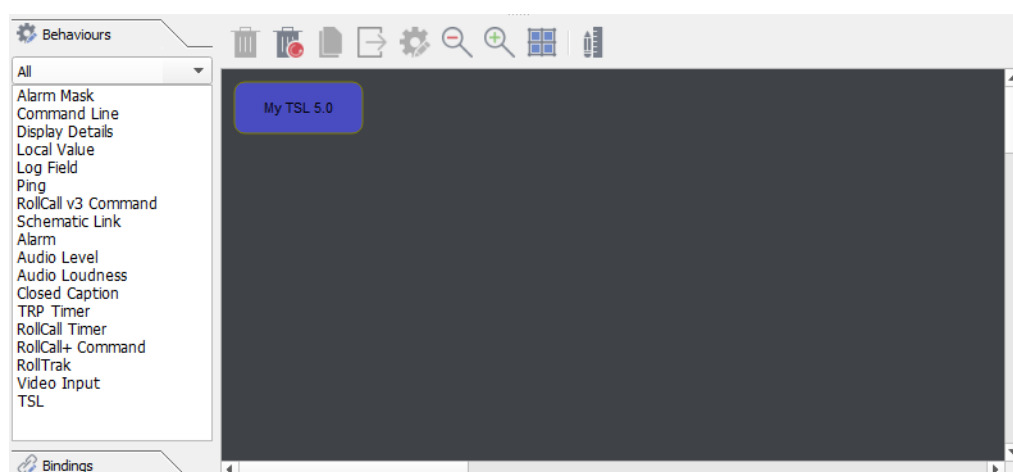


Figure 268 TSL Behaviour for the Tile itself

14. Click the **Save** icon in the **Main Menu** to save changes to the **Tile**.

A **Behaviour** has been set up on the **Tile**.

9.2.2 Lamp Widgets

Now proceed to configure the component widgets of our **DIY UMD**:

- Click on the left-hand Lamp widget in the **Tile Editor** to show the **Behaviour and Bindings Editor** for the selected widget.

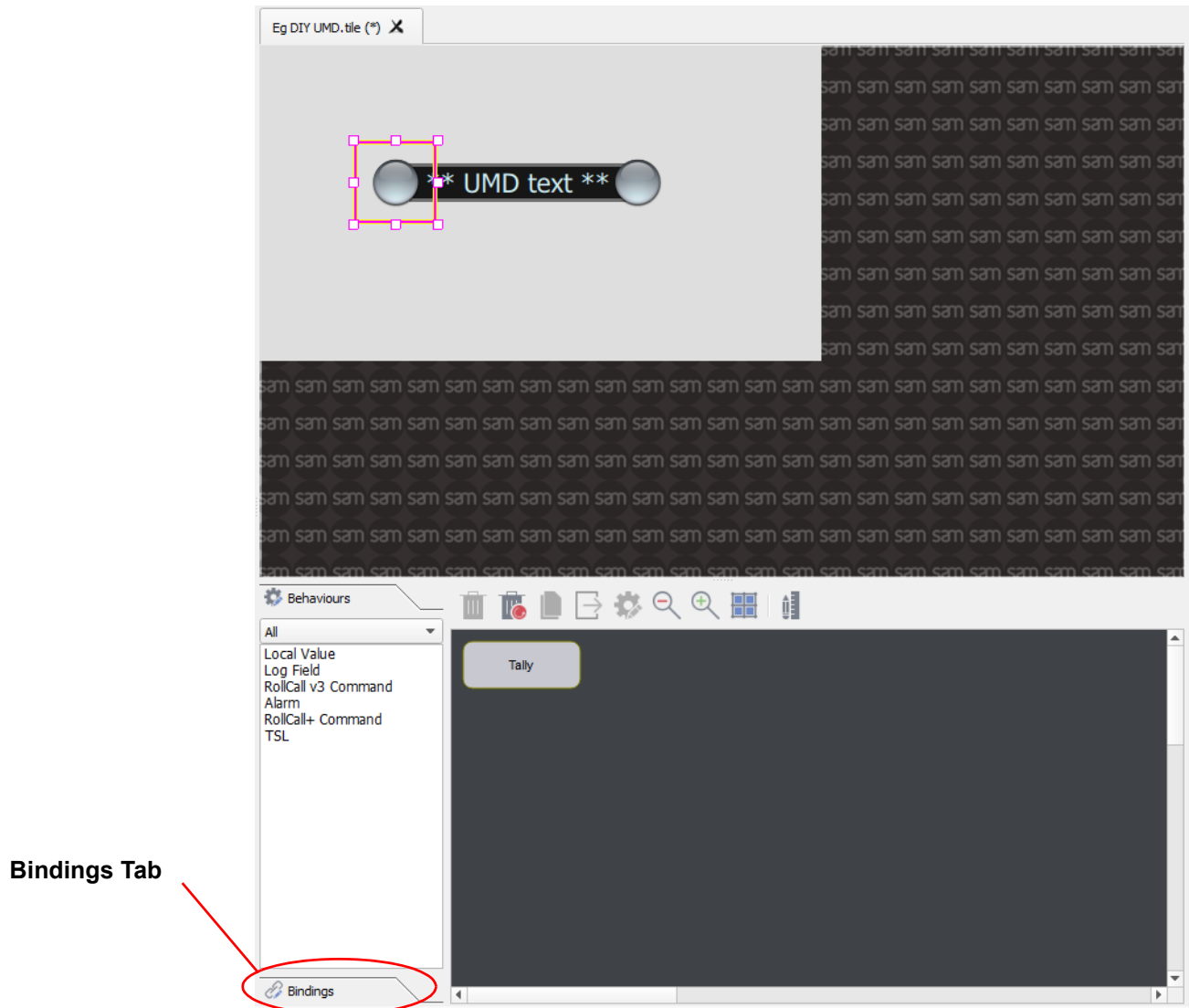


Figure 269 Behaviour and Bindings Editor for Tally Lamp Widget

- Click on the **Bindings Tab**.
- Click on the **Property** name in the list of **Bindings**.
A **Property Binding** block is added to the editor screen. See Figure 270.

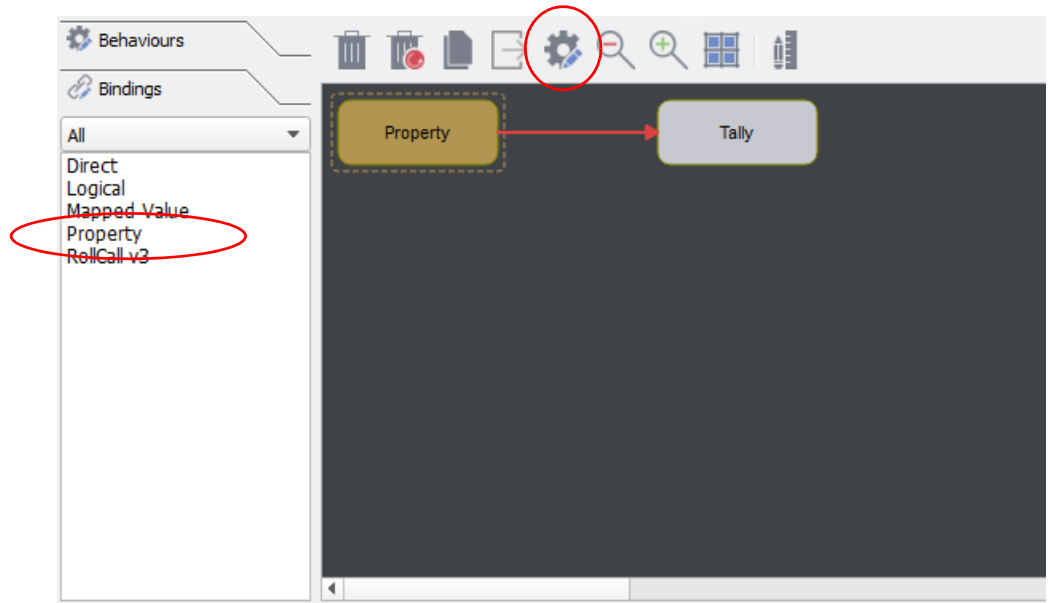


Figure 270 Added Property Binding Block

18. Select the **Property Binding** block and click on the **Open Properties** icon, see Figure 270.
Or double-click on the **Property Binding** block.
The **Properties** dialog for the **Binding** is shown.

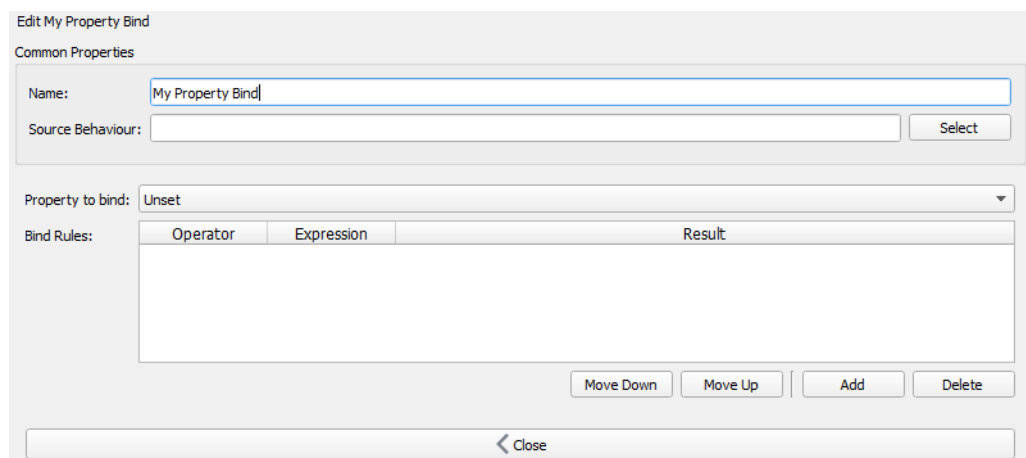


Figure 271 Property Binding Property Dialog.

19. Change the **Name** to 'My Property Bind'.
 20. Click **Select** (located to the right of **Source Behaviour**) and select 'My TSL 5.0'.
 21. At **Behaviour Value**, 'Left Lamp (0 - 3)' should already be selected. Keep this.
- This has set up which **Behaviour** data value the **Binding** will use.
22. At **Property to Bind**, select 'Tally > Lamp Color'. See Figure 272.

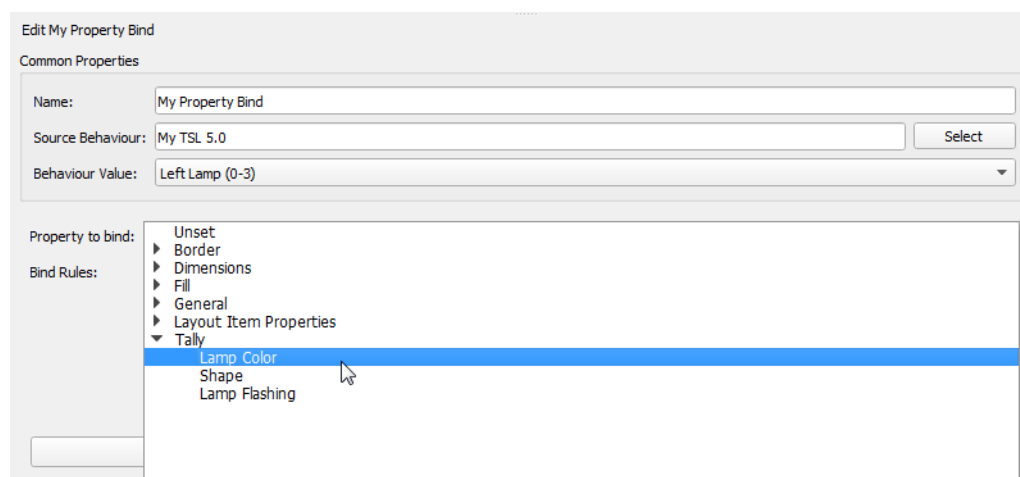


Figure 272 Property to Bind

23. At **Bind Rules**, there is an empty **Bind Rules** table. This can remain empty in this case. See Figure 273. The empty **Bind Rule** table simply passes on the **Behaviour** data value that the **Binding** is using and, in this case, this is all that is required.

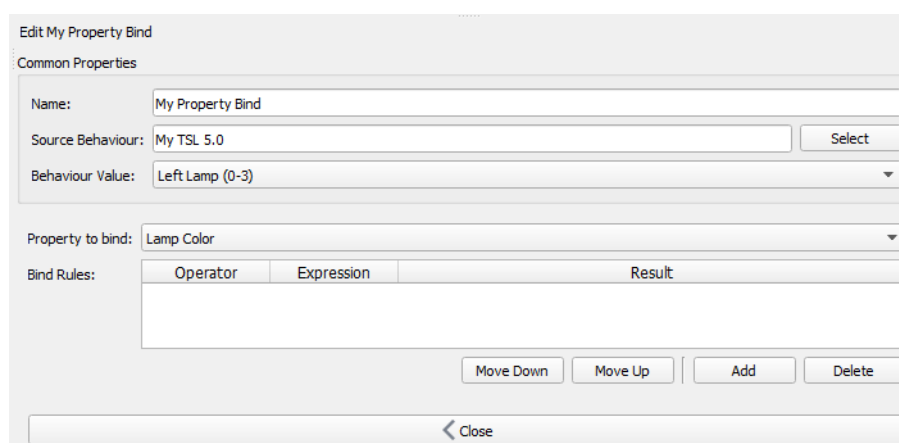
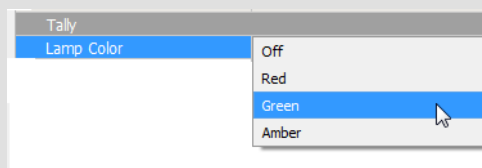


Figure 273 Bind Rule Added

24. Click **Close**.

Note:**Lamp Color and TSL Data:**

The color of a Tally Lamp widget is controlled by the widget's 'Tally > Lamp Color' property. This can be set to the values shown below, which are represented by integer values 0 to 3 respectively:



TSL data from the **TSL Behaviour** contains a 'Left Lamp' data field, which is an enum data type taking the values 0 to 3. Thus the data field can be used to set the widget property 'Tally > Lamp Color'.

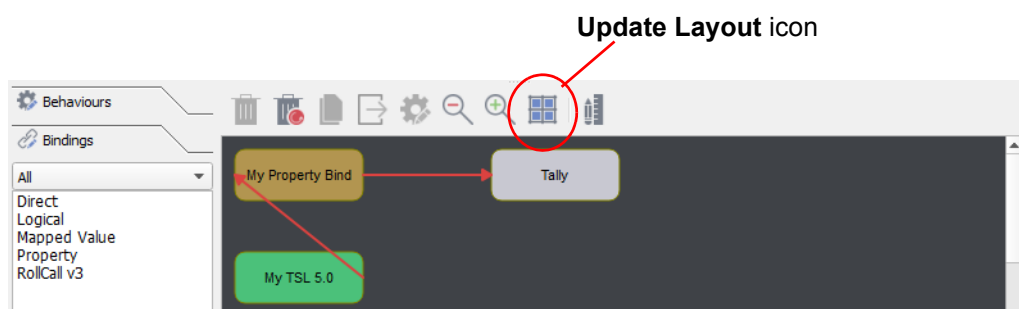


Figure 274 Tally Lamp Widget Behaviour and Binding

25. Click the **Update Layout** icon, see Figure 274, to tidy up the layout of the blocks. See Figure 275.

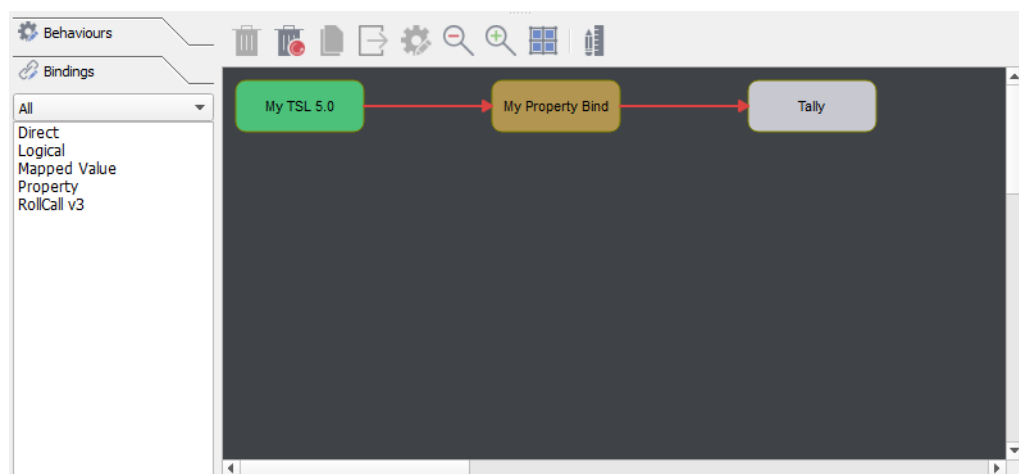


Figure 275 Tally Lamp Widget Behaviour and Binding, Updated

26. Click the **Save** icon in the **Main Menu** to save changes to the **Tile**.

The left-hand Lamp widget of the **D-I-Y UMD** has been configured.

Now proceed to configure the right-hand Lamp widget of our **DIY UMD**:

27. Click on the right-hand Lamp widget in the **Tile Editor** to show the **Behaviour and Bindings Editor** for the selected widget.

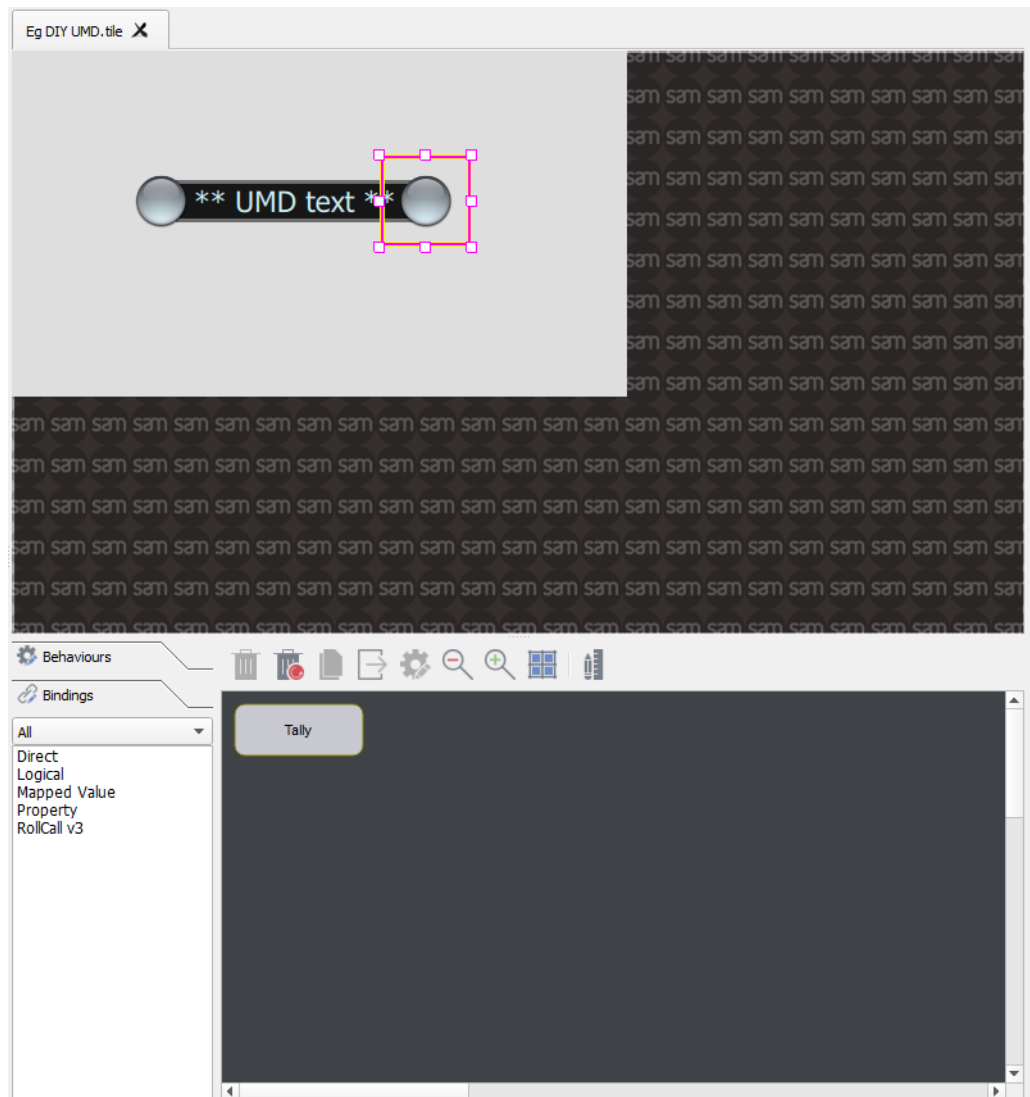


Figure 276 Behaviour and Bindings Editor for Other Tally Lamp Widget

28. For the right-hand Lamp widget, add a **Property Binding** block and configure the **Binding's** properties in a similar way to the left-hand Lamp, except:
 - Select 'Right Lamp (0 - 3)' at **Behaviour Value**.

Leave the **Bind Rule** table empty (as before, because the **Behaviour** data type value 'Right Lamp' matches the 'Tally > Lamp Color' widget property).
29. When right-hand Lamp configuration is done, close the **Binding** property dialog. The **Behaviour and Bindings Editor** screen for the right-hand Lamp should be similar to Figure 275 on page 244.
30. Click the **Save** icon in the **Main Menu** to save changes to the **Tile**.

The right-hand Lamp widget of the **D-I-Y UMD** has been configured.

9.2.3 Label Widget - Border Color

Now proceed to configure the Label widget of our **DIY UMD**:

31. Click on the Label widget in the **Tile Editor** to show the **Behaviour and Bindings Editor** for the selected widget.

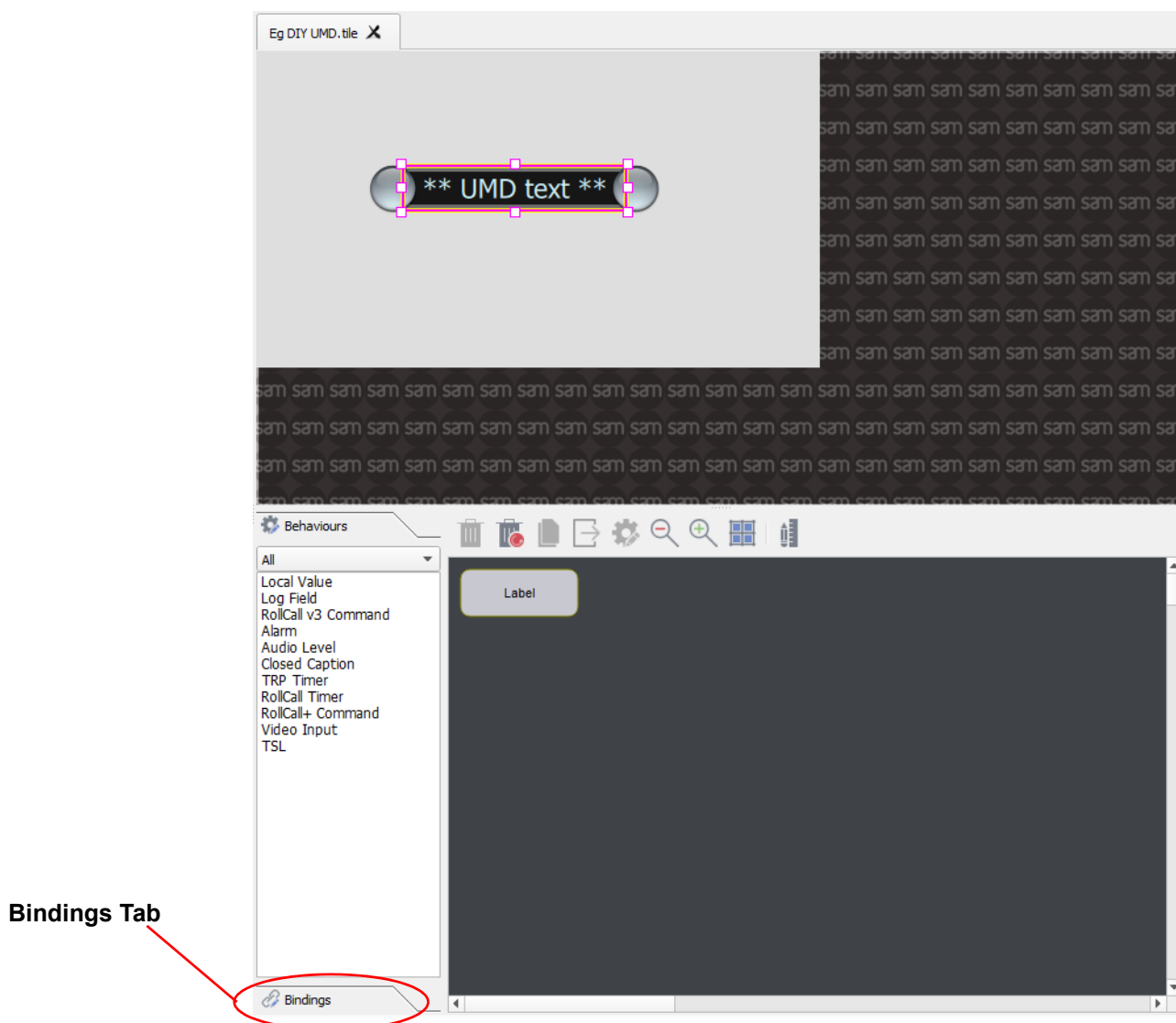


Figure 277 Behaviour and Bindings Editor for the Label Widget

32. If the list of **Behaviours** is being shown, click on the **Bindings Tab** to show the list of **Bindings**, as shown in Figure 278.

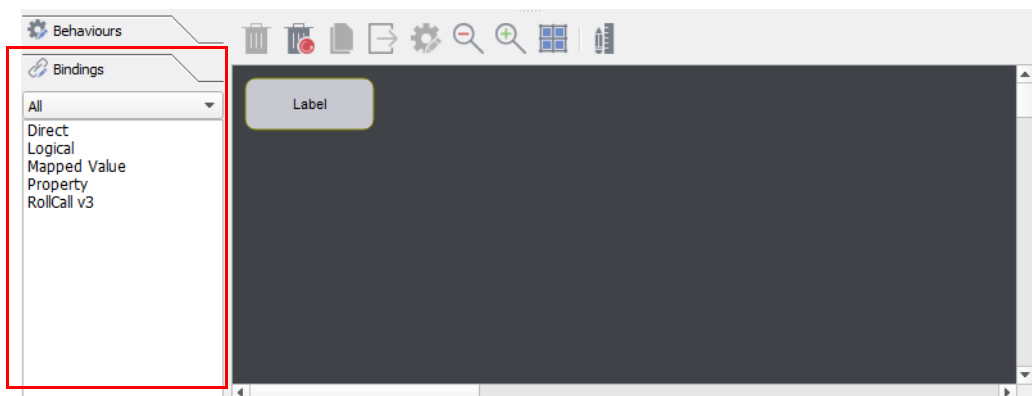


Figure 278 Showing Bindings List

33. Click on the **Property** name in the **Bindings** list.
A **Property Binding** block is added to the editor screen.
34. Add two more **Property Binding** blocks.

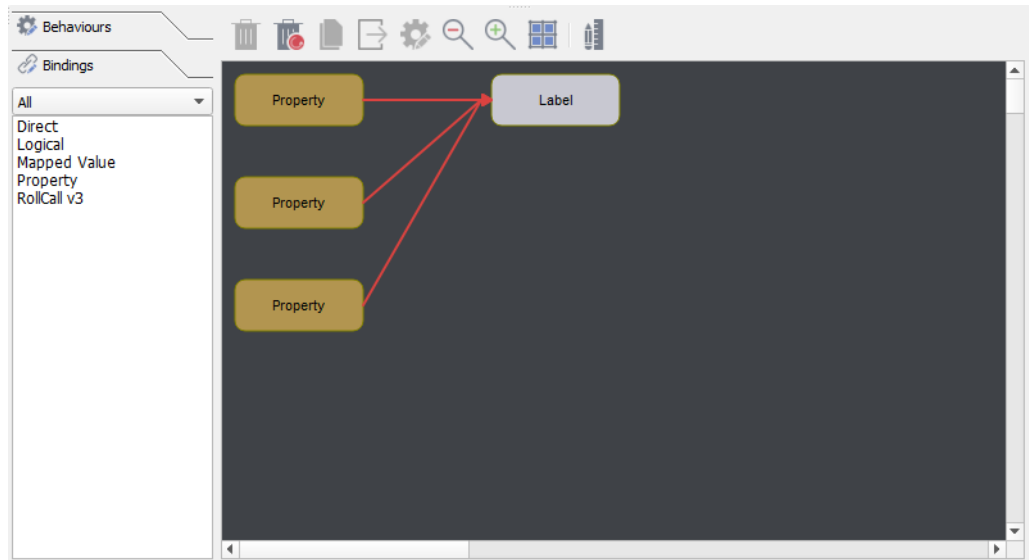


Figure 279 Binding Blocks

The first **Binding** will be used to control the Label widget's Border color:

35. Double-click on the first **Binding** block to open its **Properties** dialog.
36. Set **Name** to 'My Border Bind'.
37. At **Source Behaviour**, click **Select** and select **Behaviour** 'My TSL 5.0'. (This **Behaviour** was set up earlier.)
38. At **Behaviour Value**, select **Behaviour** data value 'Text Tally (0-3)'.

| Operator | Expression | Result |
|----------|------------|--------|
| | | |
| | | |
| | | |
| | | |

Figure 280 Property Binding Properties Dialog

39. At **Property to Bind**, select 'Border > Border Color'.

Now set up the **Bind Rules**. In this case, these set the Border Color depending on the 'Text Tally' value:

40. At **Bind Rules**, click **Add** four times.
Four empty **Bind Rule** rows are added to the **Bind Rules** table.

Edit Property

Common Properties

Name: My Border Bind

Source Behaviour: My TSL 5.0 Select

Behaviour Value: Text Tally (0-3)

Property to bind: Border Color

Bind Rules:

| Operator | Expression | Result |
|----------|------------|--------|
| Default | | %1 |
| Default | | %1 |
| Default | | %1 |
| Default | | %1 |

Move Down Move Up Add Delete

Close

Figure 281 Adding Four Empty Default Bind Rules

41. In each **Bind Rule** table row, set up:
 an Operator, an Expression and a Result.
 See Figure 283 for the final **Bind Rules** table.

Note: The Operator column is modified as shown in Figure 282.

a) Click in cell to just select.

b) Double-Click in cell to get drop-down arrow.

c) Click to get pop-up menu.

d) Click on menu item.

e) Operator modified

Figure 282 Modifying the Bind Rules Table Operator Column

Property to bind: Border Color

Bind Rules:

| Operator | Expression | Result |
|----------|------------|---------|
| Equals | 0 | #808080 |
| Equals | 1 | #800000 |
| Equals | 2 | #008000 |
| Equals | 3 | #808000 |

Move Down Move Up Add

Dark yellow Dark green Dark red Mid-gray

Figure 283 Added Four Bind Rules

42. Click **Close**.
43. Click the **Update Layout** icon.
The updated block layout is shown, see Figure 284.

Behaviour used is present at 'tile-level'

Update Layout icon

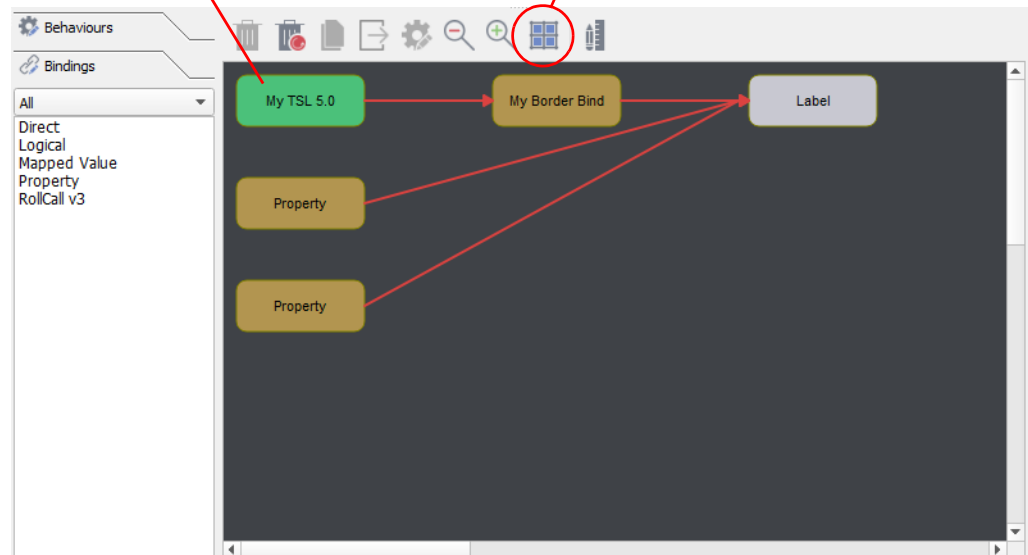


Figure 284 Label Widget, Border Property Binding

44. Click the **Save** icon in the **Main Menu** to save changes to the **Tile**.

The Label widget's border color has been configured.

9.2.4 Label Widget - Font Color

Now proceed to the Label text font color:

45. Double-click on the next **Binding** block, to open its **Properties** dialog.
46. Set **Name** to 'My Font Color Bind'.
47. At **Source Behaviour**, click **Select** and select **Behaviour** 'My TSL 5.0'.
48. At **Behaviour Value**, select **Behaviour** data value 'Text Tally (0-3)'.
49. At **Property to Bind**, select 'Font > Color'.

Now set up the **Bind Rules**. In this case, these rules will control the Font Color depending on the 'Text Tally' value:

50. At **Bind Rules**, click **Add** four times.
Four empty **Bind Rule** rows are added to the **Bind Rules** table.
51. In each **Bind Rule** table row, set up:
an Operator, an Expression and a Result.
See Figure 285 for the final **Bind Rules** table.

| Bind Rules: | Operator | Expression | |
|-------------|----------|------------|---------|
| | Equals | 0 | #ffffff |
| | Equals | 1 | #ff8080 |
| | Equals | 2 | #80ff80 |
| | Equals | 3 | #ffff80 |
| | | | |

Light yellow 
 Light green 
 Light red 
 White

Figure 285 Added Four Bind Rules

52. Click **Close**.
53. Click the **Update Layout** icon.
The block layout is tidied up (and the final Binding block may need moving slightly to look like the block layout is shown in Figure 284.

Behaviour used is present at 'tile-level'

Update Layout icon

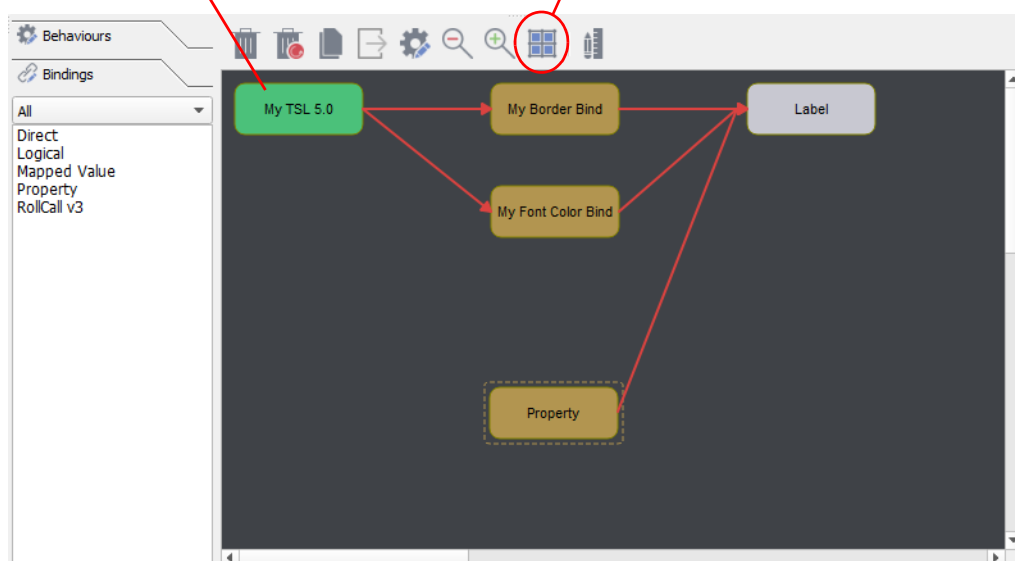


Figure 286 Label Widget, Border and Font Color Binding

54. Click the **Save** icon in the **Main Menu** to save changes to the **Tile**.
The Label widget's font color has been configured.

9.2.5 Label Widget - Caption Text

Now proceed to the Label caption text:

55. Double-click on the third **Binding** block, to open its **Properties** dialog.
56. Set **Name** to 'My Caption Bind'.
57. At **Source Behaviour**, click **Select** and select **Behaviour** 'My TSL 5.0'.
58. At **Behaviour Value**, select **Behaviour** data value 'Label'.
59. At **Property to Bind**, select 'General > Caption'.

Now set up the **Bind Rules**. In this case, these rules will provide the Caption text:

60. At **Bind Rules**, there is an empty **Bind Rules** table.
This can remain empty in this case. See Figure 287.

The empty **Bind Rule** table simply passes on the **Behaviour** data value that the **Binding** is using and, in this case, this is all that is required.
This is because the **Behaviour** data value is 'Label' of the TSL Data, which is an string data type, and this matches the widget property being bound to (General > Caption).

Figure 287 shows the 'Edit My Caption Bind' dialog box. The 'Common Properties' section contains the following fields:

- Name:** My Caption Bind
- Source Behaviour:** My TSL 5.0 (with a 'Select' button)
- Behaviour Value:** Label (with a dropdown arrow)
- Property to bind:** Caption (with a dropdown arrow)

The 'Bind Rules' section contains an empty table with the following columns:

| Operator | Expression | Result |
|----------|------------|--------|
| | | |

At the bottom of the dialog are buttons for 'Move Down', 'Move Up', 'Add', 'Delete', and a 'Close' button.

Figure 287 Label Widget, Caption Binding Properties

61. Click **Close**.
62. Click the **Save** icon in the **Main Menu** to save changes to the **Tile**.

The Label widget's caption text has been configured.

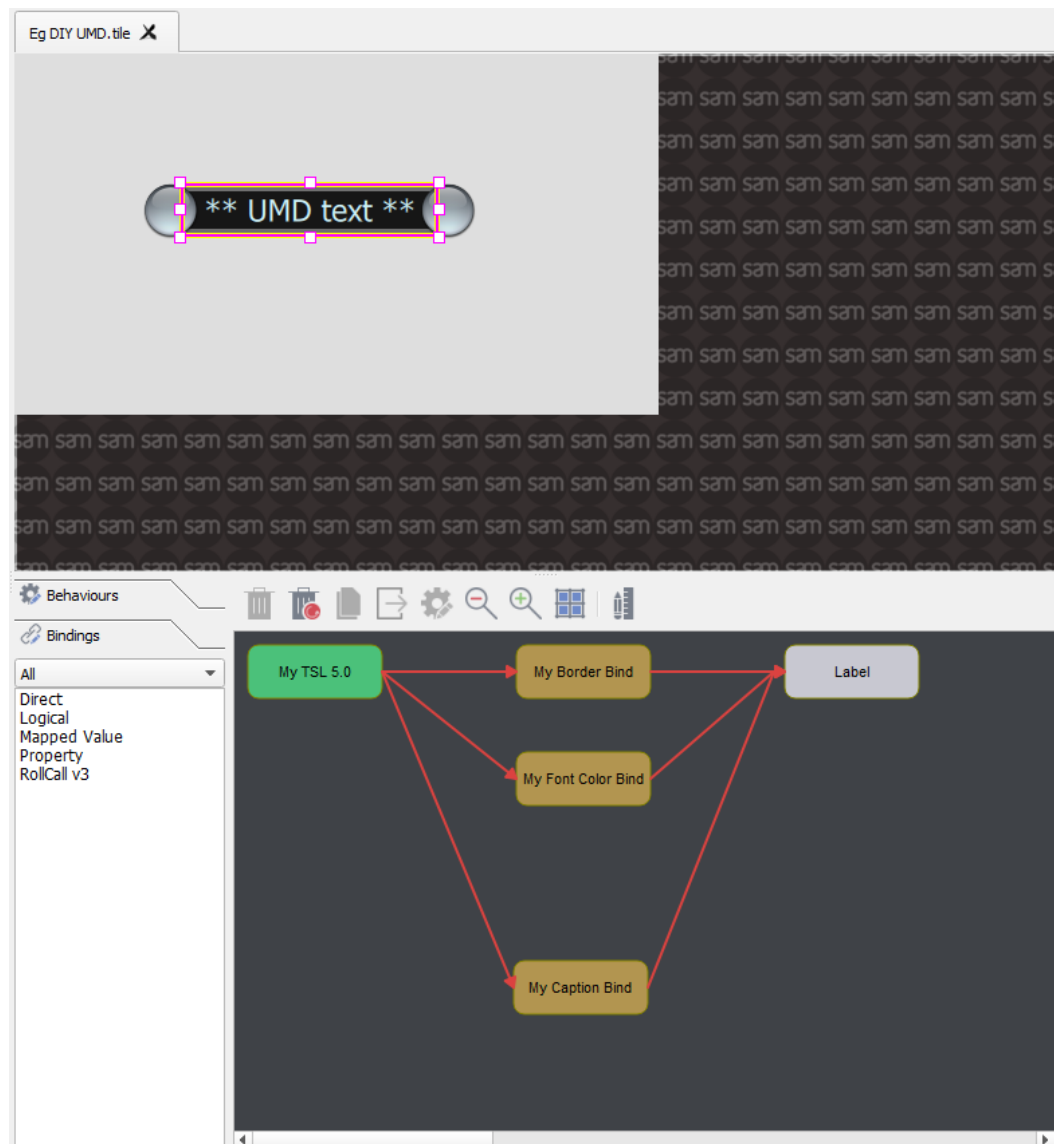


Figure 288 Label Widget, Behaviours and Bindings

The **Behaviours** and **Bindings** have been configured. Figure 289 shows the Orbit screen.

Save icon

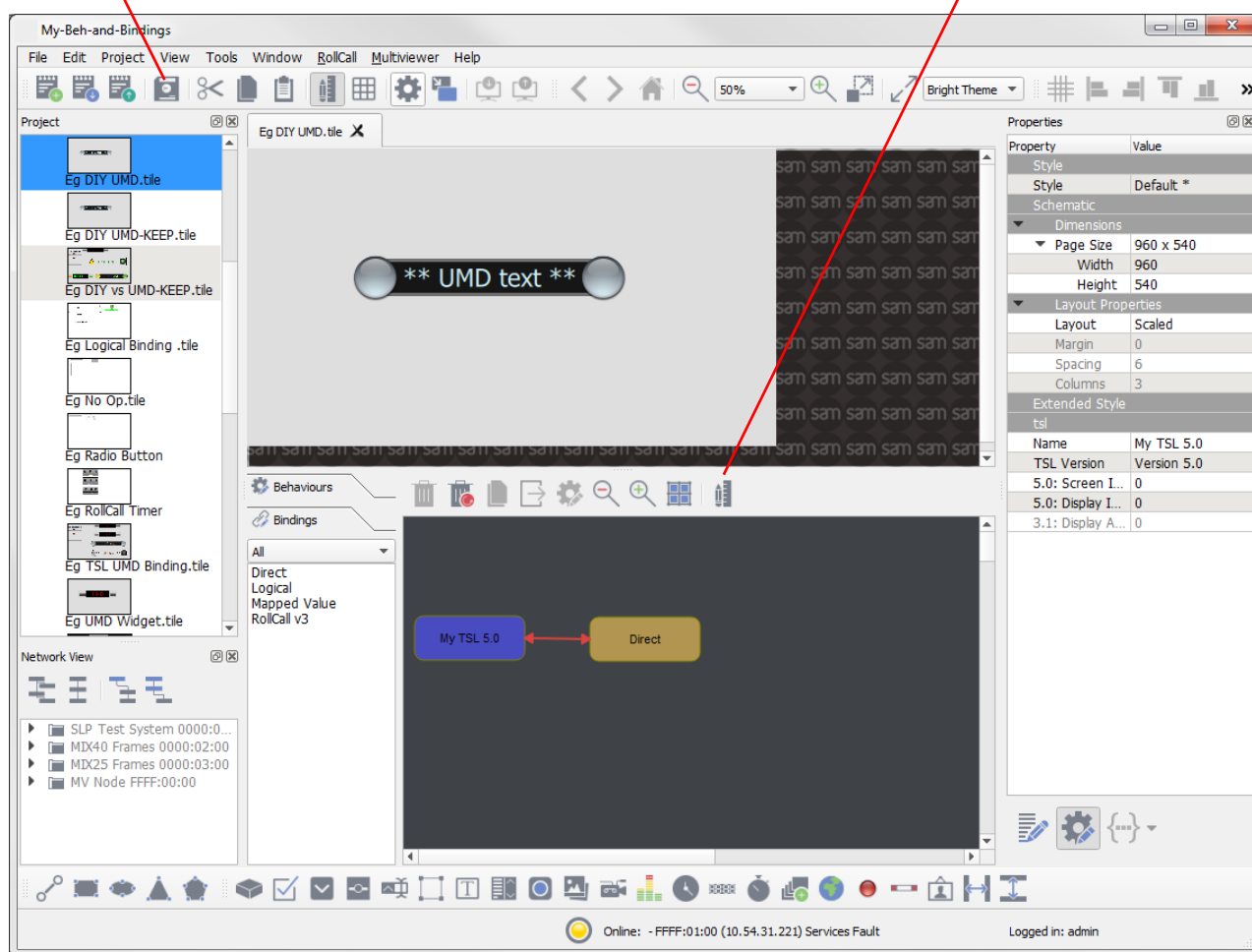
Behaviour and Bindings **Run Mode** icon

Figure 289 Completed D-I-Y UMD

Configuration for the D-I-Y UMD is done.

Click the Behaviours and Bindings **Run Mode** icon to exercise the widget. See Section 9.6 “Exercise the Tile and Widgets” on page 283.

9.2.6 D-I-Y UMD Behaviour and Bindings Recap

The **TSL Behaviour** (a TSL Listener) gets data from a TSL server in the system. The **TSL Behaviour** presents information about a certain video feed to the widgets that make up the **D-I-Y UMD**. Figure 290 summarizes the Behaviour and Bindings set up.

TSL Behaviour at 'tile-level'

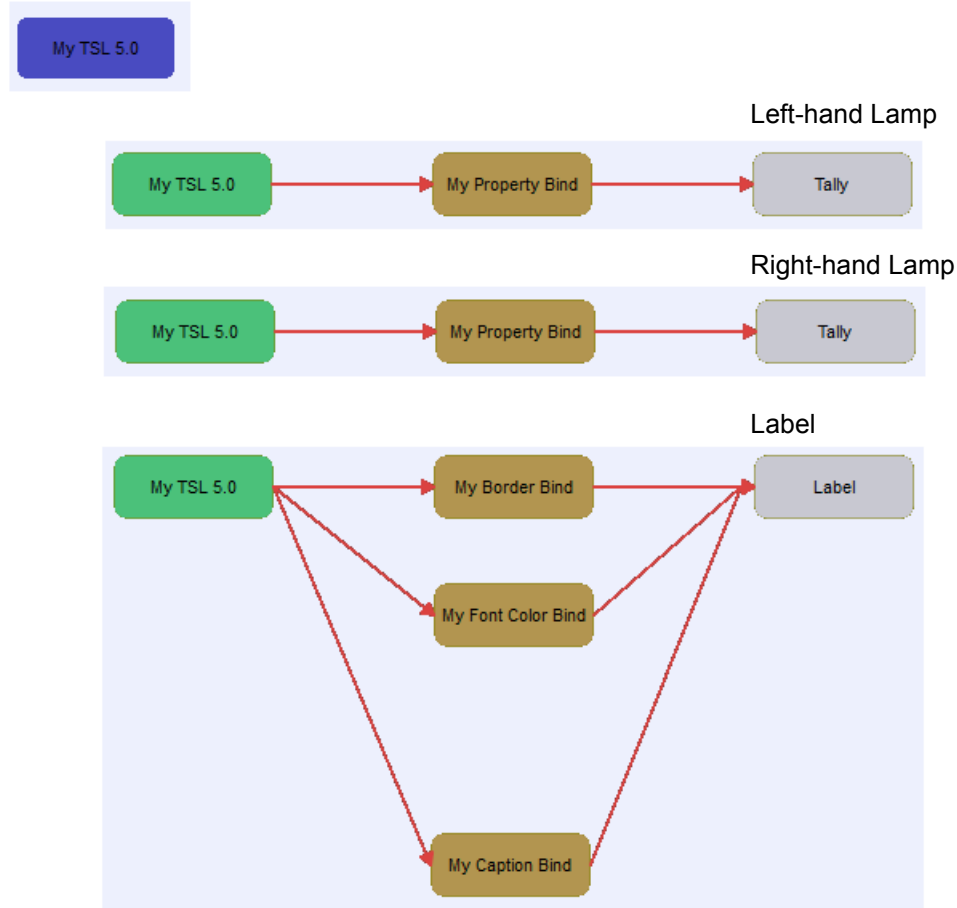


Figure 290 'D-I-Y UMD, All Widget Behaviours and Bindings

9.3 Another Do-It-Yourself UMD (Shape Fill and Images)

A customized UMD can be built from separate, individual widgets and configured to have the required functionality. Various widget types may be employed to show the UMD information.

This example uses:

- A **TSL Behaviour**.
- **Property Bindings**.
- Shape, Image and Label widgets.

Figure 291 compares a UMD widget to the example UMD made from widgets.

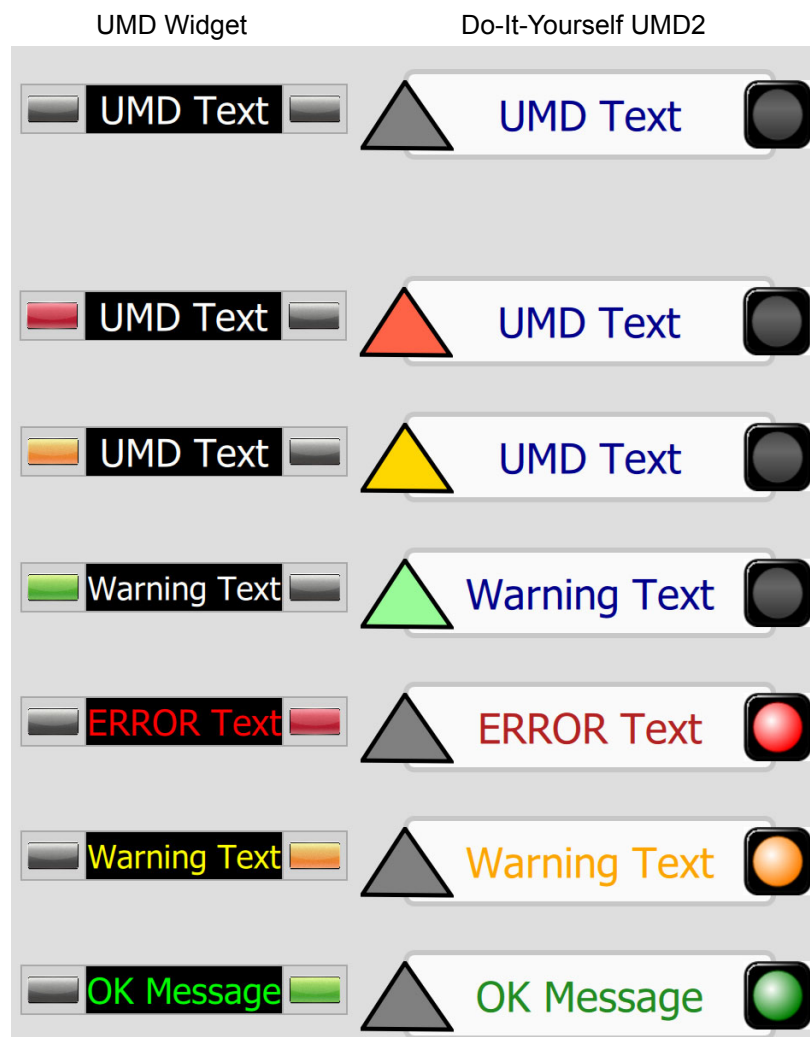


Figure 291 'Comparing UMD Widget and Do-It-Yourself UMD 2 Operation

The example **D-I-Y UMD 2** comprises a triangle shape, an image and a Label widget, as shown in Figure 292.

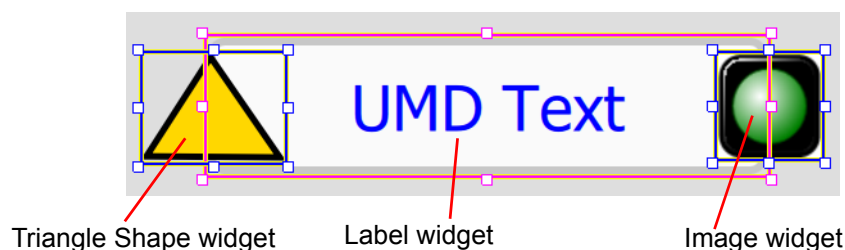


Figure 292 'Widgets of Do-It-Yourself UMD 2

TSL Protocol Version 5.0 will be used in this example and each component widget will be set up to react to some TSL system data:

- **Left-hand Tally Shape** - To change color according to the TSL 'Left Lamp' data value.
- **Right-hand Tally LED Image** - To change color according to the TSL 'Right Lamp' data value.
- **Label** widget - This is required to: Show the TSL Label text data, and to change font color according to the TSL 'Text Tally' data value.

The required **Behaviour** and **Binding** connectivity is shown in Figure 293.

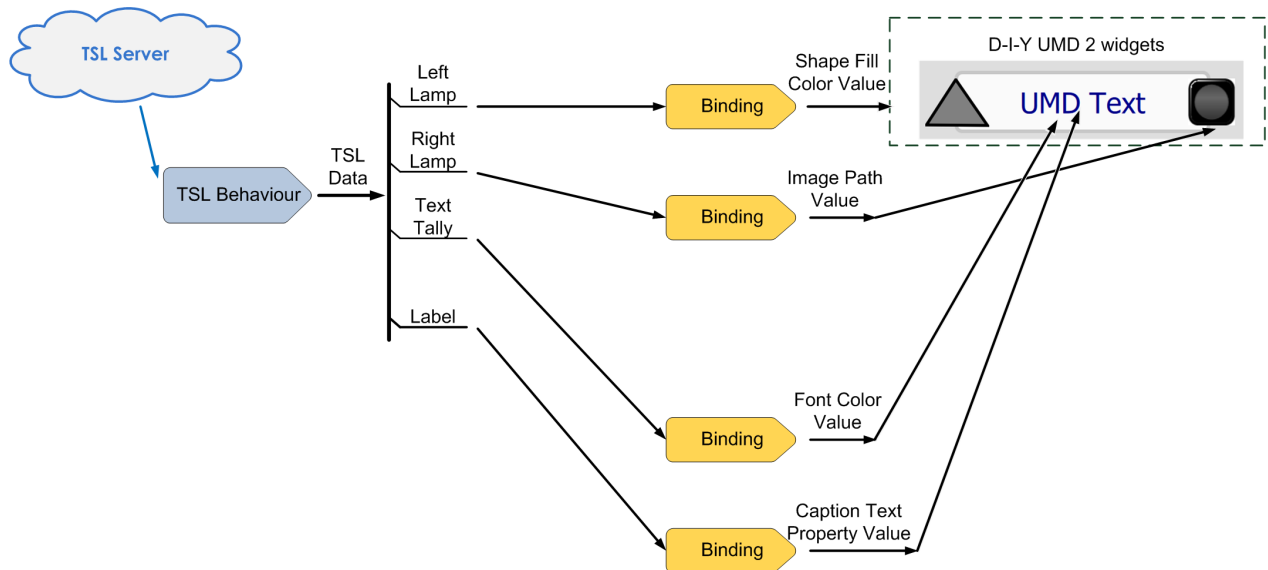


Figure 293 Connectivity for Do-It-Yourself UMD 2

The TSL Behaviour's TSL data is used several times by the widgets and is common to them.

9.3.1 Configuring Behaviours and Bindings

A description of how to configure the functionality of this **D-I-Y UMD 2** using **Behaviours** and **Bindings** follows:

1. Create a new empty tile, for example, a quarter-sized tile, and form a UMD from a shape widget, an image widget and a Label widget. See Figure 292.
2. Open the graphical **Behaviours and Bindings Editor** by clicking the **Edit Behaviours** icon in the Tile Editor's Advanced Tool Bar.



3. Click on the **Tile** to *deselect* any widgets. This has, by default, selected the **Tile** itself. The **Behaviours** and **Bindings** screen for the **Tile** itself is shown. There are no **Behaviours** nor **Bindings** set up yet. See Figure 294.

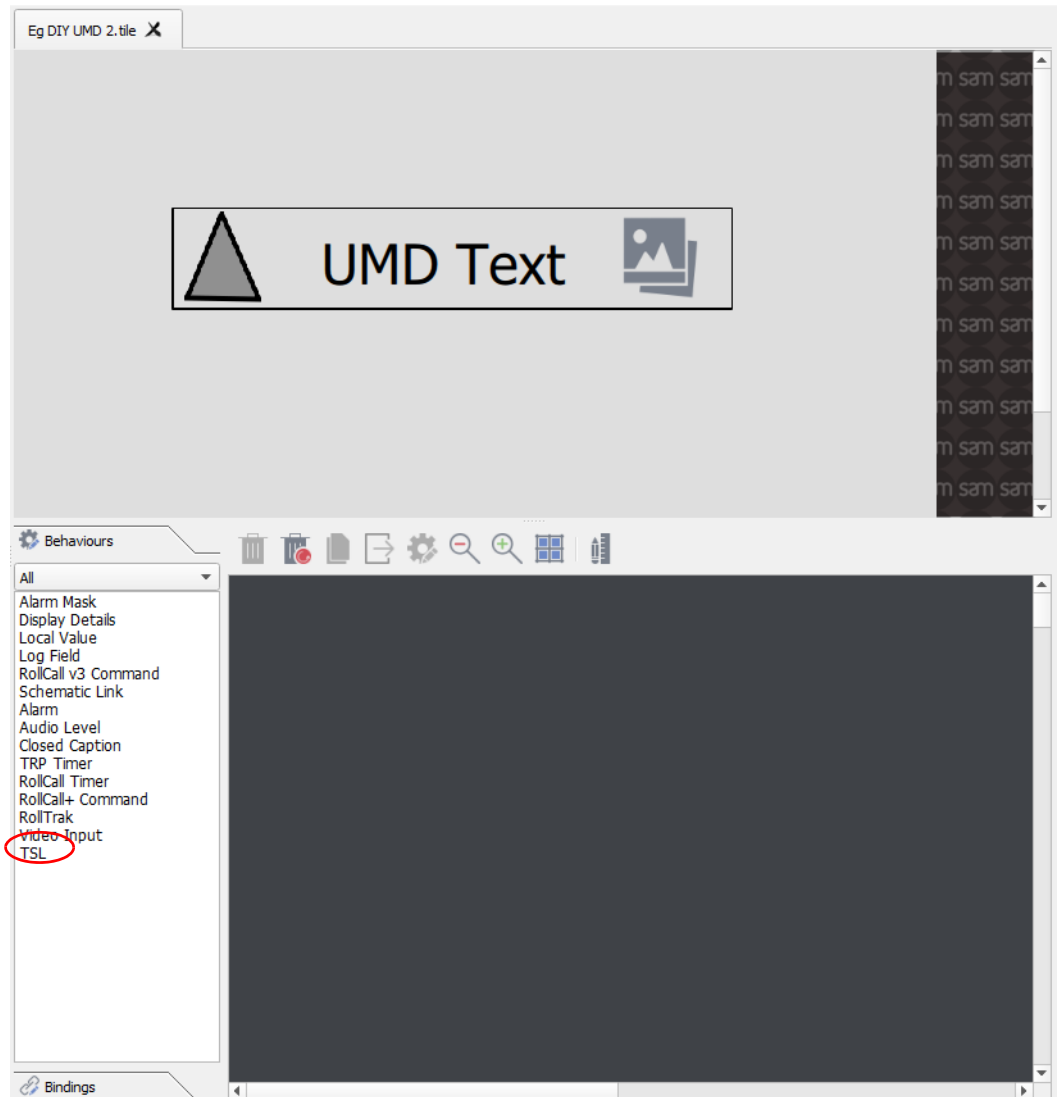


Figure 294 Behaviour and Bindings Editor

Initially, we add a **Behaviour** and a **Binding** to the **Tile**, at 'tile-level' (similar to **DIY UMD** in Section 9.2) before adding to each component widget.

Add a **TSL Behaviour**:

4. Click on the **TSL** name in the list of **Behaviours**.
A **TSL Behaviour** block is added to the editor screen.
5. Select the **TSL Behaviour** block and click on the **Open Properties** icon.
Or double-click on the **TSL Behaviour** block.
To open the **Properties** dialog for the **Behaviour**.

Edit the **Behaviour's** properties (similar to **D-I-Y UMD** in Section 9.2.1):

6. Change property **Name** to 'My TSL 5.0'.
7. Change property **TSL Version** to 'Version 5.0'.
8. Select property **5.0 Screen Index**.
Click the **Select Variable** 'down-arrow' icon. (See Figure 264 on page 239.)
Select 'Create Variable from Property'.
Click **OK**.

9. Select property **5.0 Display Index**.
Click the **Select Variable** 'down-arrow' icon.
Select 'Create Variable from Property'.
Click **OK**.
10. Click **Close**.
11. Click the **Save** icon in the **Main Menu** to save changes to the **Tile**.

The **TSL Behaviour** properties have been set up on the **Tile**.
(Similar to **DIY UMD**, see Figure 267 on page 240 for the completed properties dialog.)

Now proceed to configure the component widgets of **DIY UMD 2**.

9.3.2 Shape Widget

Now proceed to configure the left-hand "lamp" of our **DIY UMD 2**:

12. Click on the **Shape** widget in the **Tile Editor** to show the **Behaviour and Bindings Editor** for the selected widget.

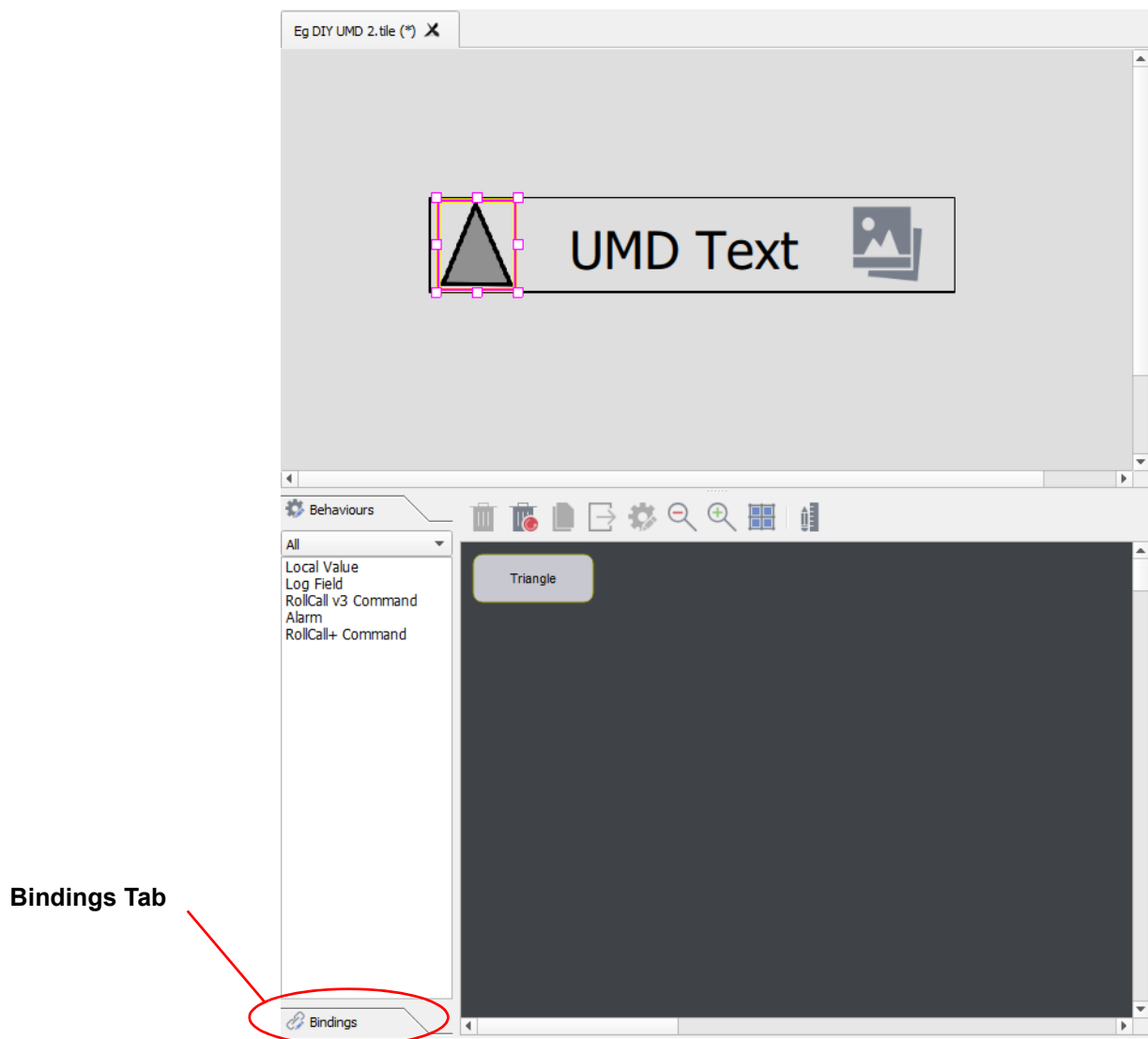


Figure 295 Behaviour and Bindings Editor for Shape Widget

13. Click on the **Bindings Tab**.

14. Click on the **Property** name in the list of Bindings.
A **Property Binding** block is added to the editor screen.

Edit the **Binding**'s properties:

15. Double-click on the **Property Binding** block.
The **Properties** dialog for the **Binding** is shown.
16. Change the **Name** to 'My Property Bind'.
17. Click **Select** at **Source Behaviour** and select the 'My TSL 5.0' **Behaviour**.
18. At **Behaviour Value**, 'Left Lamp (0 - 3)' should already be selected. Keep this.

This has set up which **Behaviour** data value for the **Binding** to use.

19. At **Property to Bind**, select 'Fill > Fill Color'. See Figure 296.

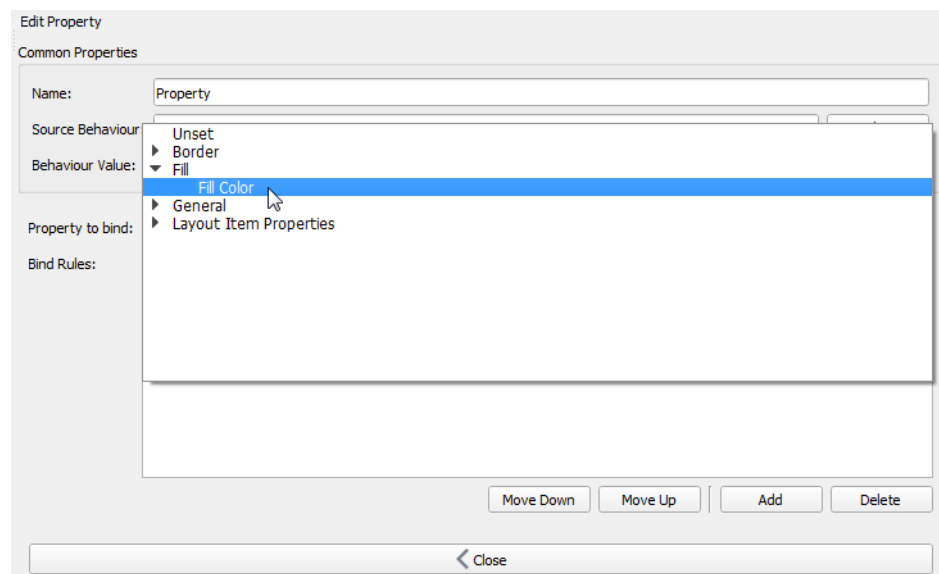


Figure 296 Property to Bind

20. At **Bind Rules**, click **Add** four times.
Four new **Bind Rule** rows are added to the **Bind Rules** table. See Figure 297.

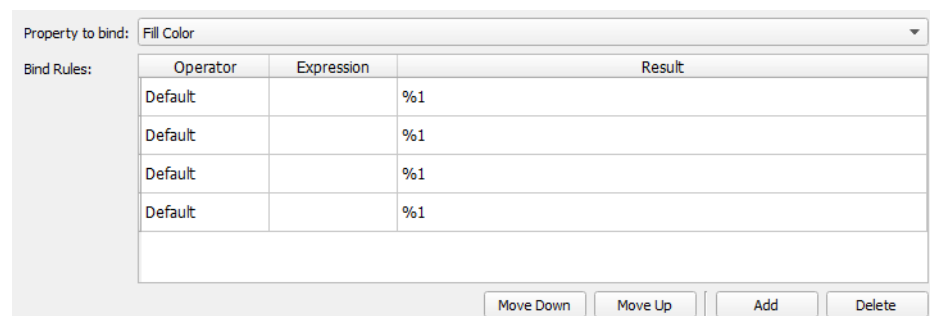


Figure 297 Default Bind Rule Added

The default **Bind Rules** added simply passes on the **Behaviour** data value that the **Binding** is using and, for our previous **D-I-Y UMD** example, this was all that was required.

But, for our current **D-I-Y UMD 2** example here, we need to generate a color value result for the target 'Fill Color' widget property.

Now enter the **Bind Rules**:

21. Change the **Bind Rules** table to map the incoming **Behaviour** data value (Values 0, 1, 2 and 3) to colors.
The completed set of **Bind Rules** is given in Figure 298.

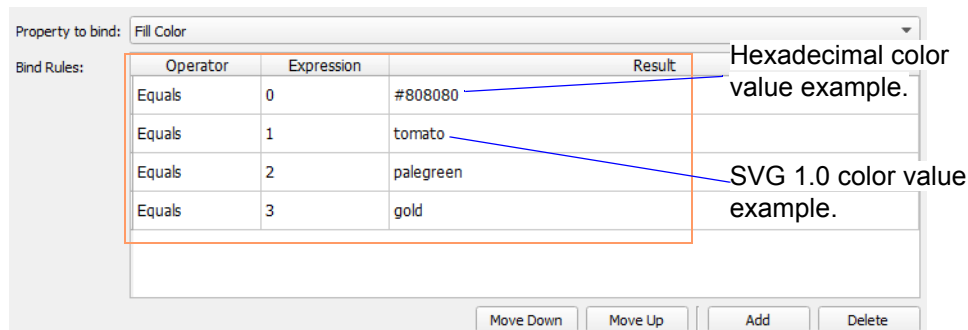


Figure 298 Bind Rules Added

22. Click **Close**.
23. Click the **Update Layout** icon to tidy up the layout of the blocks.
Figure 299 shows the resulting **Behaviour and Binding Editor** view.

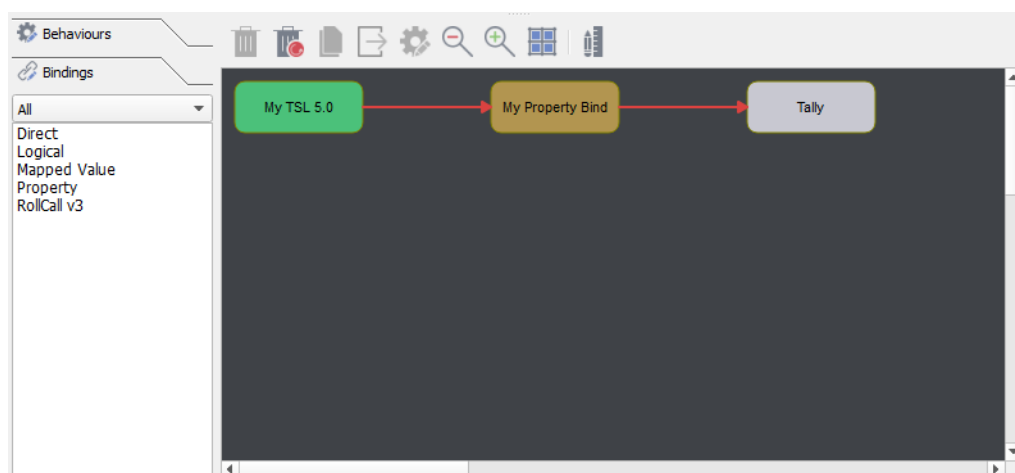


Figure 299 Tally Lamp Widget Behaviour and Binding, Updated

24. Click the **Save** icon in the **Main Menu** to save changes to the **Tile**.
- The left-hand “lamp” of the **D-I-Y UMD 2** has been configured.

9.3.3 Image Widget

Now proceed to configure the right-hand “lamp” of our **DIY UMD 2**:

25. Click on the **Image** widget in the **Tile Editor** to show the **Behaviour and Bindings Editor** for the selected widget.

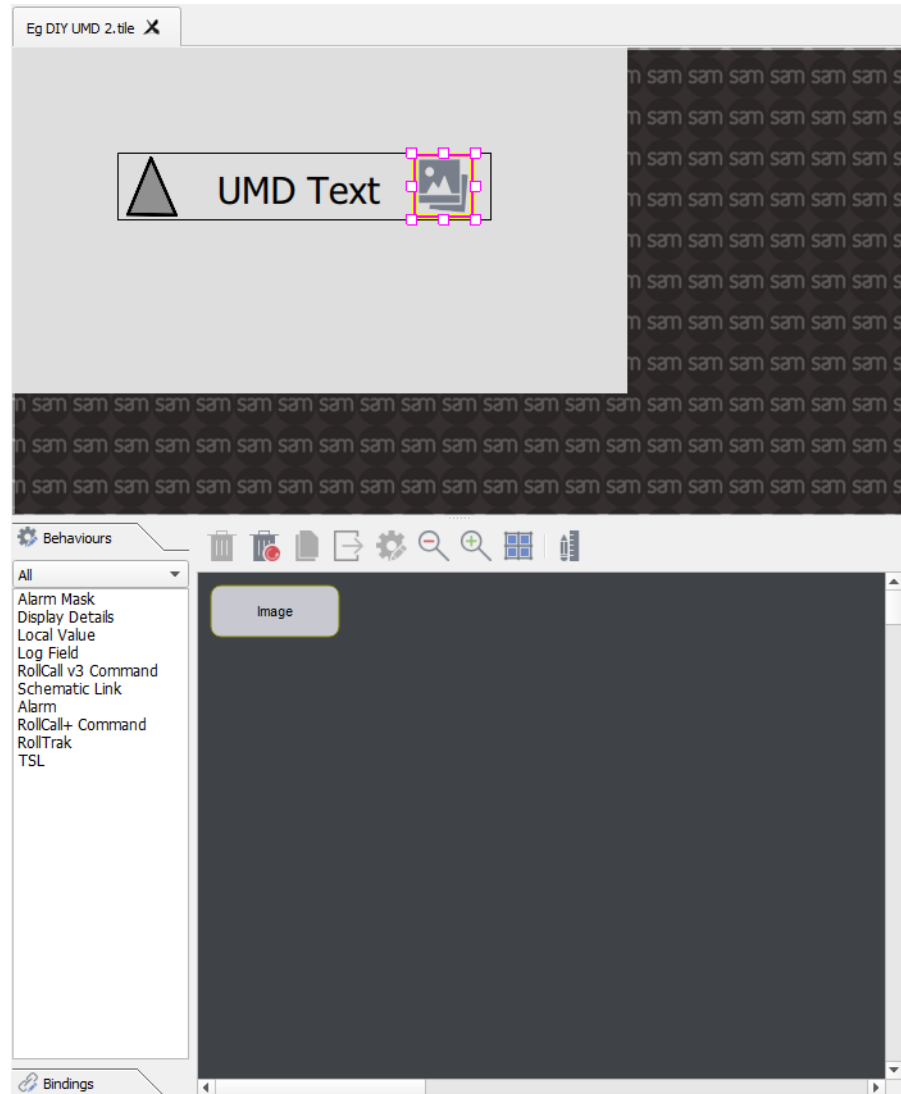


Figure 300 Behaviour and Bindings Editor for Image Widget

26. Add a **Property Binding** block.

Edit the **Binding**'s properties:

27. Double-click on the **Property Binding** block.
The **Properties** dialog for the **Binding** is shown.
28. Change the **Name** to 'My Property Bind'.
29. Click **Select** at **Source Behaviour** and select the 'My TSL 5.0' **Behaviour**.
30. At **Behaviour Value**, select 'Right Lamp (0 - 3)'.

Now select which widget property to bind to:

31. At **Property to Bind**, select 'Image > Source Path'.

Next, some **Bind Rules** will be set up to control which image is shown by our **Image** widget.

The four images we are using here have been imported into the Orbit project. They are present as image resources, shown in Figure 301.

In the **Project View**, expand the 'resources > images' folder to show the image resources in the Orbit project.

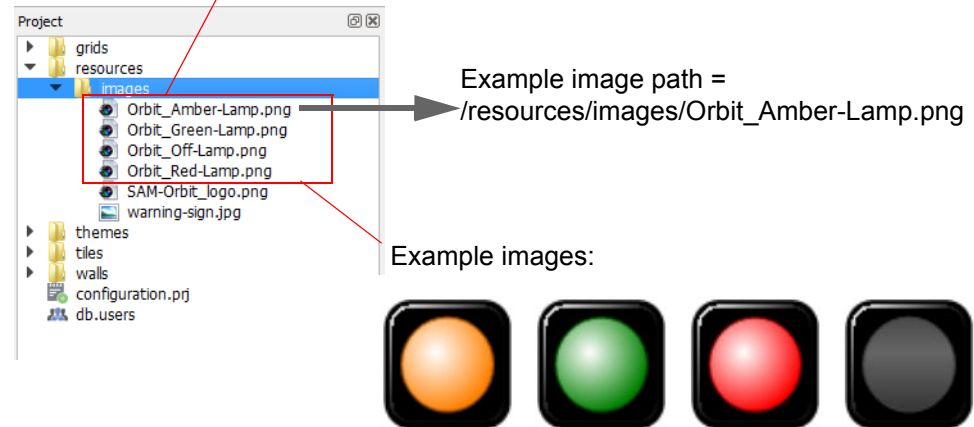


Figure 301 Four LED Indicator Images

Now enter the **Bind Rules** to produce a result for our 'Image > Source Path' widget property:

32. At **Bind Rules**, click **Add** four times.
Four new **Bind Rule** rows are added to the **Bind Rules** table.
33. Change the **Bind Rules** table to map the incoming **Behaviour** data value (Values 0, 1, 2 and 3) to be image paths.
The completed set of **Bind Rules** is given in Figure 302.

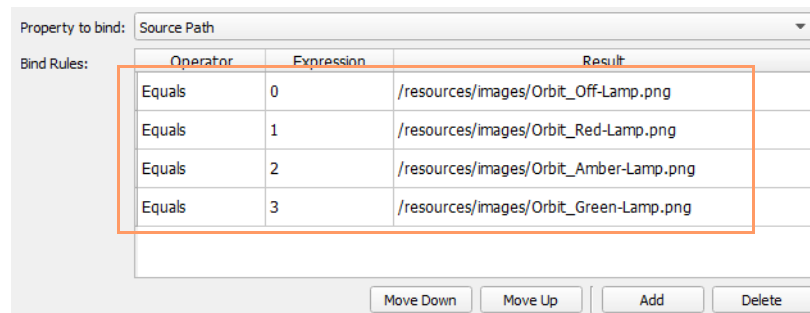


Figure 302 Bind Rules Added

34. Click **Close**.
35. Click the **Save** icon in the **Main Menu** to save changes to the **Tile**.

The right-hand "lamp" of **D-I-Y UMD 2** has been configured.

9.3.4 Label Widget

36. Configure the **Label** widget of our **DIY UMD 2** in a similar way to the **DIY UMD**, see Section 9.2.3 on page 246, Section 9.2.4 and Section 9.2.5.
37. Click **Close**.
38. Click the **Save** icon in the **Main Menu** to save changes to the **Tile**.

The label of **D-I-Y UMD 2** has been configured.

9.3.5 D-I-Y UMD Behaviour and Bindings Recap

The **TSL Behaviour** (a TSL Listener) gets data from a TSL server in the system. The **TSL Behaviour** presents information about a certain video feed to the widgets that make up the **D-I-Y UMD 2**. Figure 303 summarizes the Behaviour and Bindings set up.

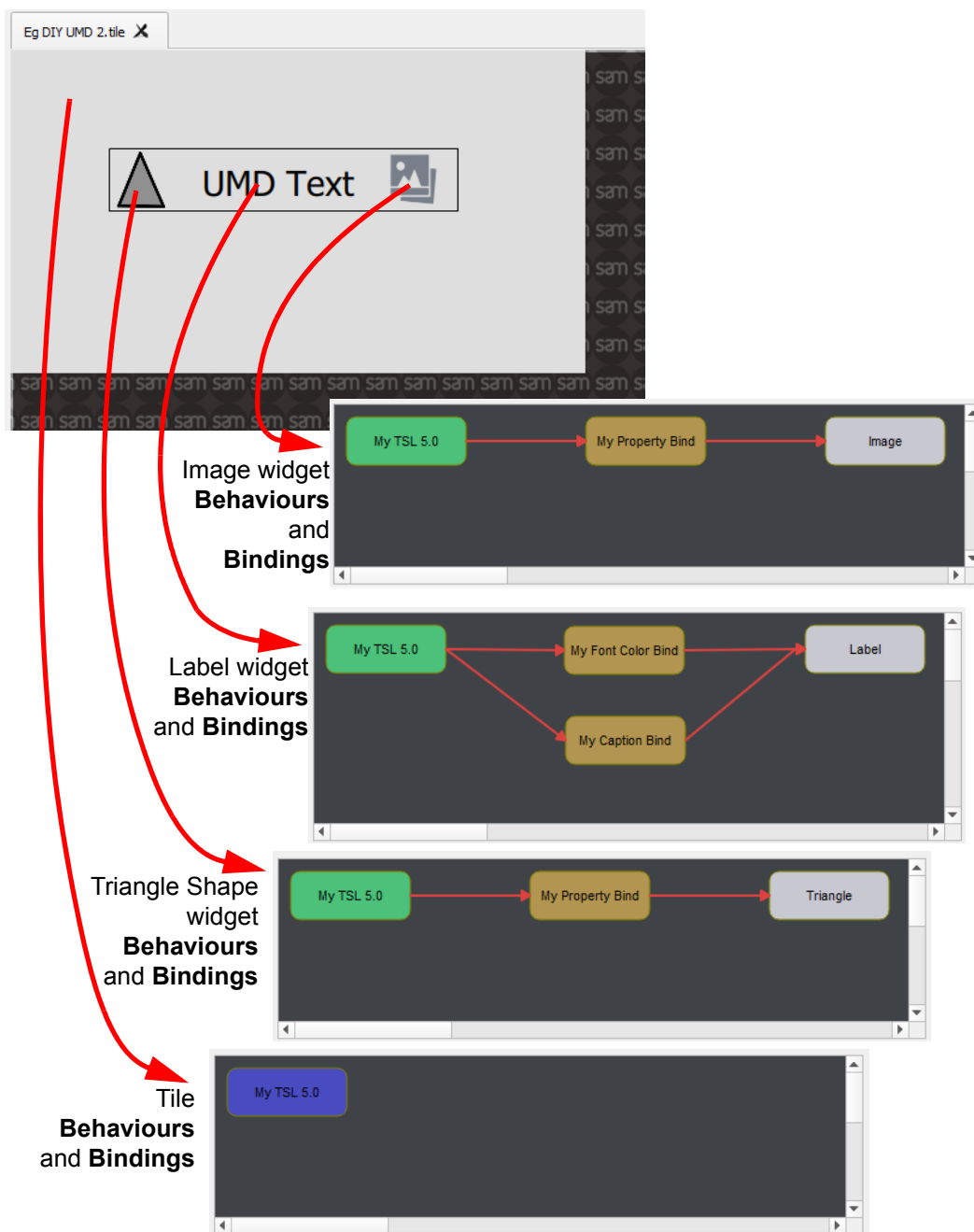


Figure 303 D-I-Y UMD 2 Behaviours and Bindings

9.4 Border Tally Example

In some cases, a customized behavior is required to show some UMD information. For example, a border changing color instead of a lamp changing color on-screen with a text overlay which switches on and off.

The best starting point for building such a tile for multiviewer applications is a **Video Auto-Tile**.

This example uses:

- A **TSL Behaviour**.
- **Property Bindings**.
- A **Rectangle Shape** widget and a **Label** widget.

Figure 304 shows such a tile.

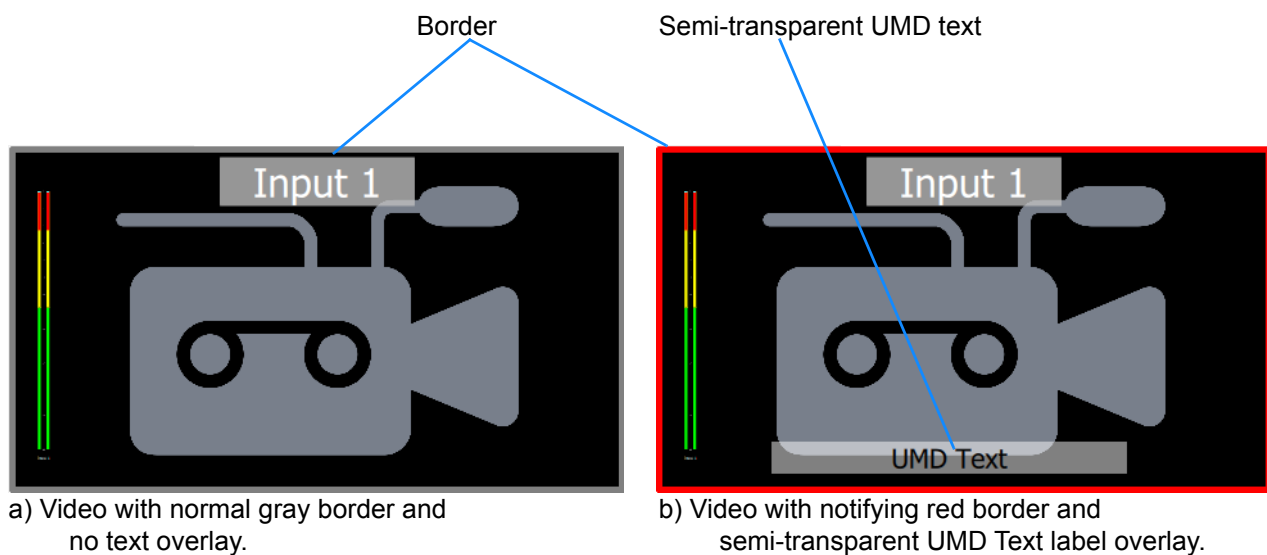


Figure 304 Example Border Tally Tile:

- Video widget with gray border, no overlay text.
- Video widget with red border and overlay text.

The example **Border Tally** tile has an *additional* **Rectangle Shape** widget and a **Label** widget, see Figure 305.

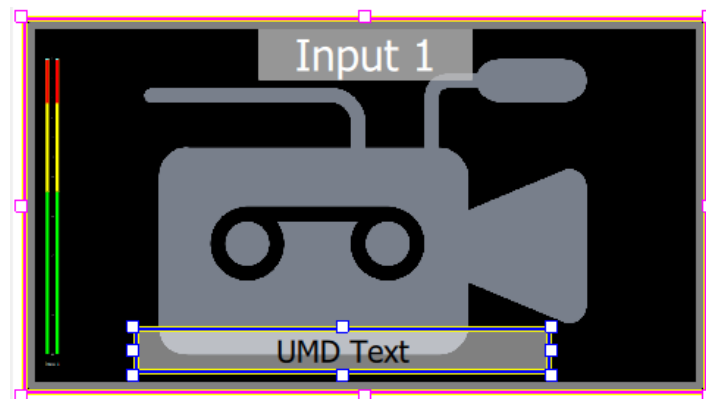


Figure 305 Border Tally Rectangle and Label

TSL Protocol Version 5.0 will be used in this example and each component widget will be set up to react to some TSL system data:

- **Rectangular** shape border - To change color according to the TSL 'Left Lamp' data value.
- **Label** widget - Normally invisible, and then show the TSL Label text data on a semi-transparent background when the Text Tally value is asserted.

The required **Behaviour** and **Binding** connectivity is shown in Figure 306.

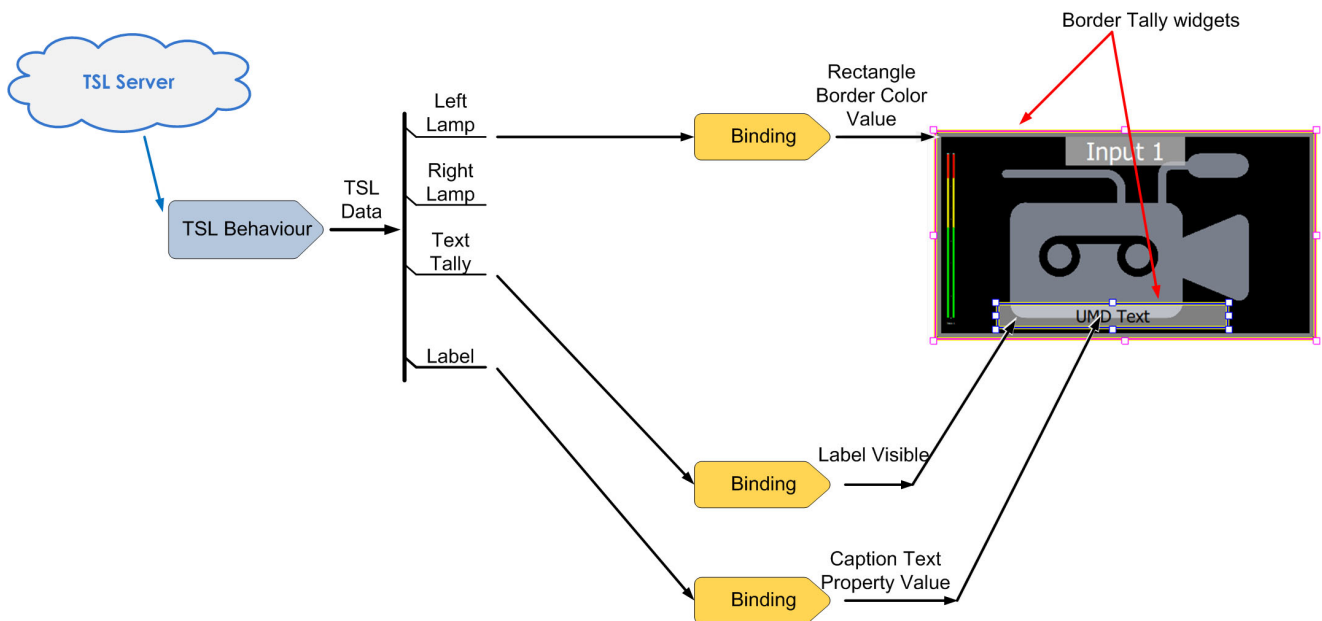


Figure 306 Border Tally Behaviour and Binding Connectivity

The TSL Behaviour's TSL data is used several times by the widgets and is common to them.

9.4.1 Configuring Behaviours and Bindings

A description of how to configure the functionality for a **Border Tally** using **Behaviours** and **Bindings** follows:

1. Create a new **Custom Tile** from a video **Auto-Tile** and open it in the **Tile Editor**. (The example here has a tile size of 1920 x 1080.)

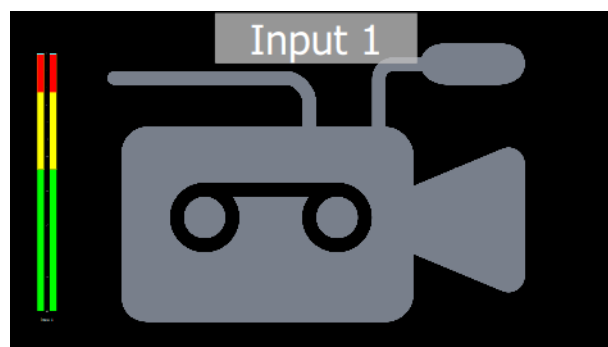
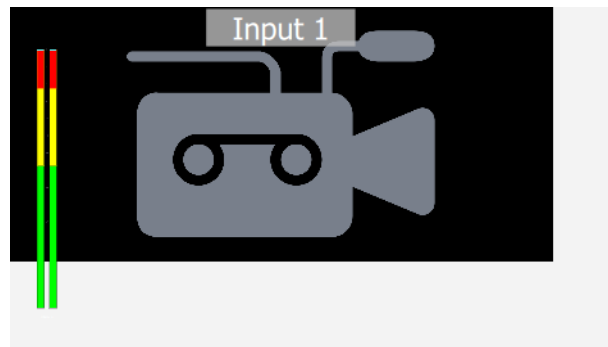


Figure 307 Video Auto-Tile

The **Rectangle** and **Label** widgets require adding to the **Tile**:

2. In order to show the added widgets and ease widget selection on the tile, temporarily reduce the size of the video widget, as shown in Figure 308.

Figure 308 Video Widget *Temporarily Resized (Smaller)*

3. Add a **Rectangle** widget and a **Label** widget, see Figure 309.

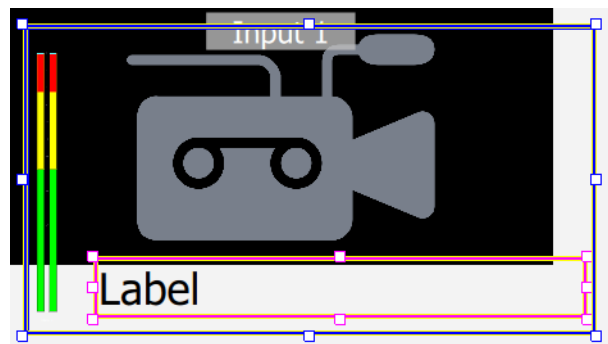
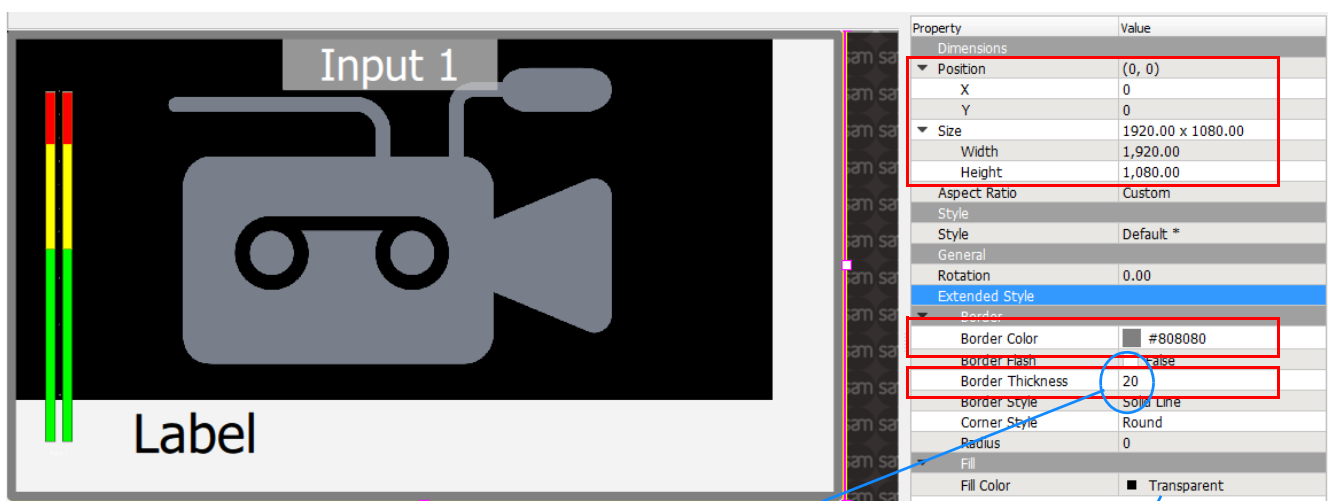


Figure 309 Added Rectangle and Label Widgets

Now set up some widget properties:

4. Select the **Rectangle** widget and modify its properties. See Figure 310.

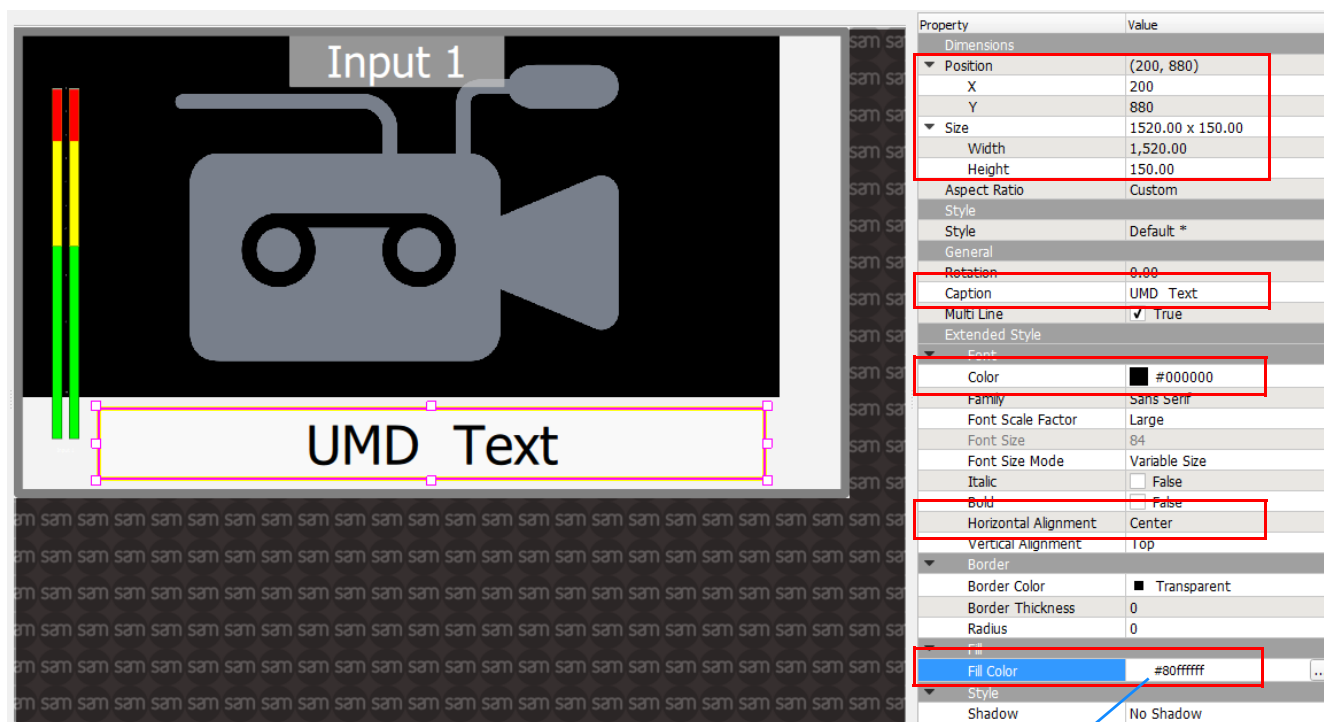


Note: Border Thickness

Note: No Fill Color (Fill is fully transparent)

Figure 310 Modified Rectangle

5. Select the **Label** widget and modify its properties. See Figure 311.



Note: Translucent Fill (Fill 50% transparent)

Figure 311 Modified Label

The widgets have been set up. The widget functionality must now be defined:

9.4.1.1 Add a Behaviour to the Tile Itself

6. Click on the **Tile Editor** background to select the **Tile** itself, no widgets. And click the **Edit Behaviours** icon. See Figure 312. The **Tile's** blank **Behaviour and Binding Editor** screen is shown.

Click in the **Tile Editor** background area to select the **Tile** itself.

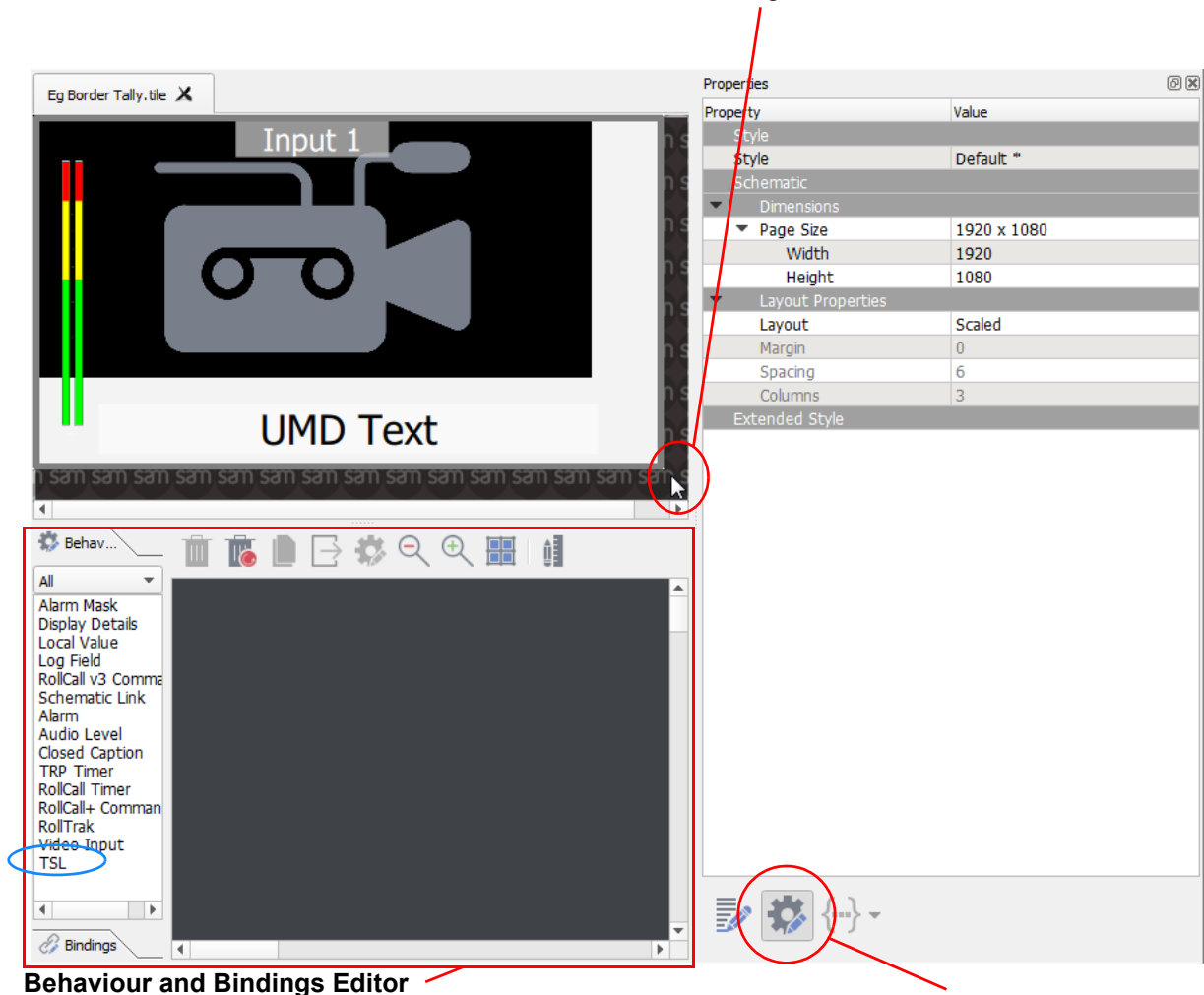


Figure 312 Tile's Blank Behaviour and Bindings Editor

Initially, a **Behaviour** is added to the **Tile**, at 'tile-level', before other component widgets:

7. Click on the **TSL** name in the list of **Behaviours**. A **TSL Behaviour** block is added to the editor screen.

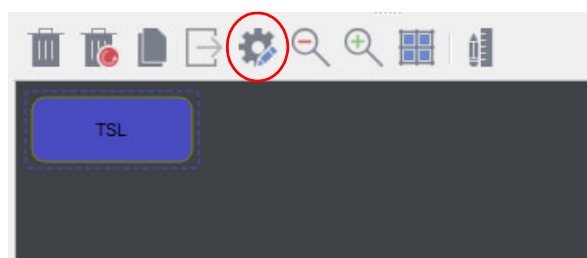


Figure 313 TSL Behaviour Block

8. Select the **TSL Behaviour** block and click on the **Open Properties** icon, see Figure 313.

Or double-click on the **TSL Behaviour** block.
The **Properties** dialog for the **Behaviour** is shown.

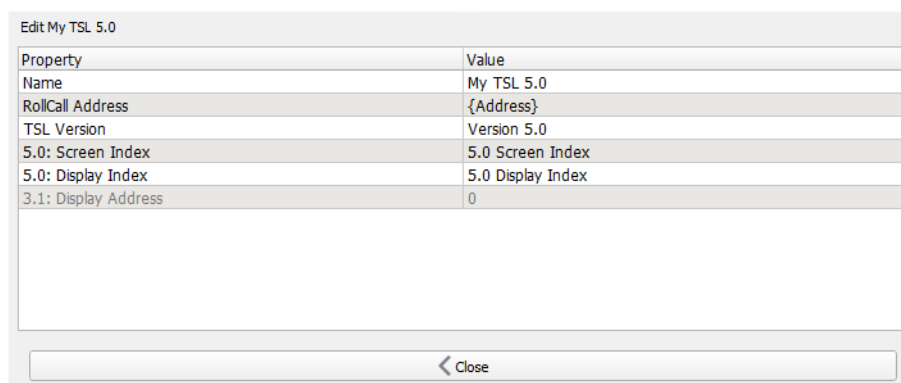
9. Change property **Name** to 'My TSL 5.0'.
10. Change property **TSL Version** to 'Version 5.0'.

The other two TSL version 5.0 **Binding** properties, '5.0 Screen Index' and '5.0 Display Index', may now be edited. Each is initially set to a fixed number, '0'.

In order that the property values can be changed each time the tile is used, a system variable may be created and used for each property value:

11. Select property **5.0 Screen Index**.
Click the **Select Variable** 'down-arrow' icon.
And select 'Create Variable from Property'.
12. Click **OK**.
13. Select property **5.0 Display Index**.
Click the **Select Variable** 'down-arrow' icon.
And select 'Create Variable from Property'.
14. Click **OK**.

The **TSL Behaviour** properties have been set up, see Figure 314 for the completed properties dialog.



| Property | Value |
|----------------------|-------------------|
| Name | My TSL 5.0 |
| RollCall Address | {Address} |
| TSL Version | Version 5.0 |
| 5.0: Screen Index | 5.0 Screen Index |
| 5.0: Display Index | 5.0 Display Index |
| 3.1: Display Address | 0 |

Close

Figure 314 Completed TSL Behaviour Properties Dialog

15. Click **Close**.

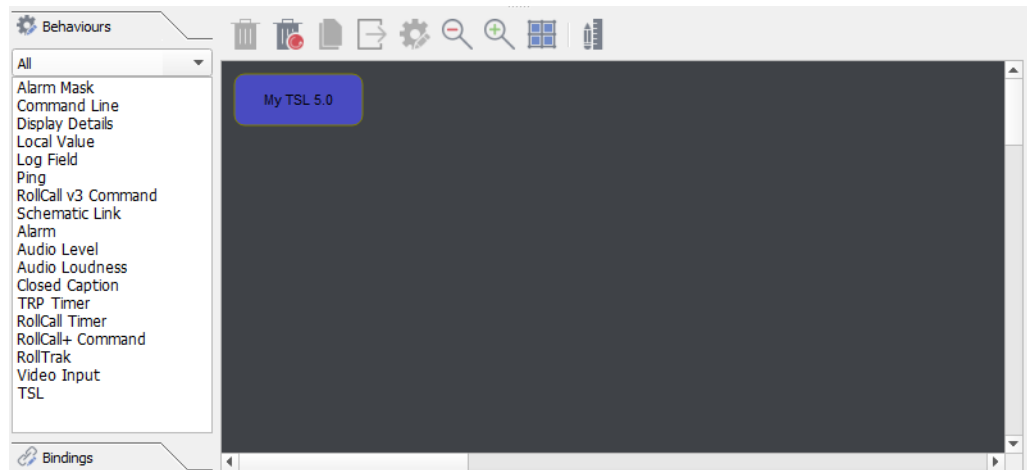


Figure 315 TSL Behaviour for Tile

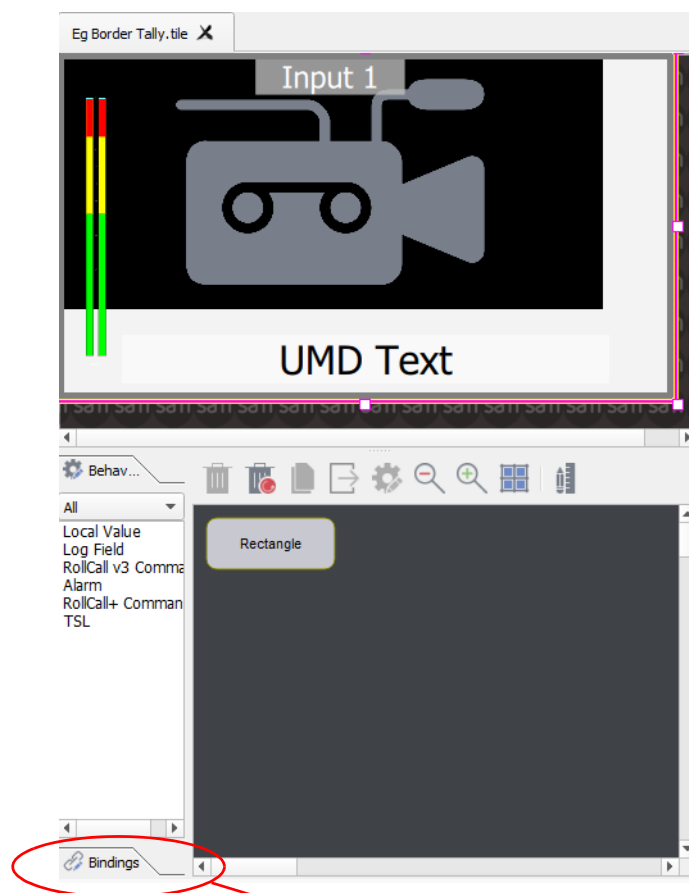
16. Click the **Save** icon in the **Main Menu** to save changes to the **Tile**.

A **Behaviour** has been set up on the **Tile**.

Now proceed to configure the relevant widgets.

9.4.2 Border Behaviour and Binding

17. Select the **Rectangle** widget in the **Tile Editor** to show the **Behaviour and Bindings Editor** for the selected widget.



Bindings Tab

Figure 316 Behaviour and Bindings Editor for Rectangle Widget

18. Click on the **Bindings** Tab.
19. Click on the **Property** name in the list of **Bindings**.
A **Property Binding** block is added to the editor screen. See Figure 317.

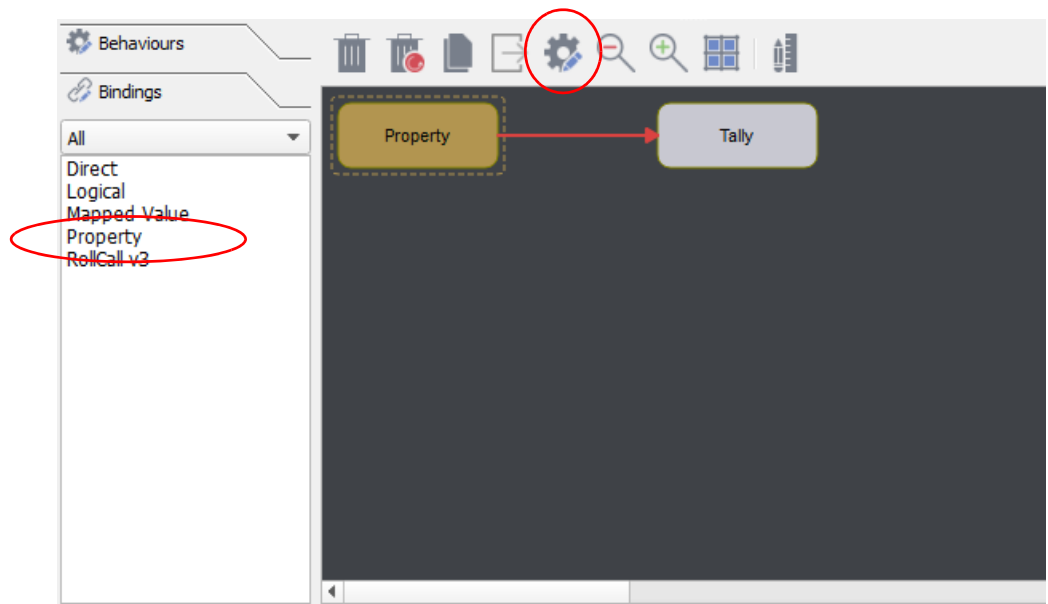


Figure 317 Added Property Binding Block

20. Double-click on the **Property Binding** block.
The **Properties** dialog for the **Binding** is shown.

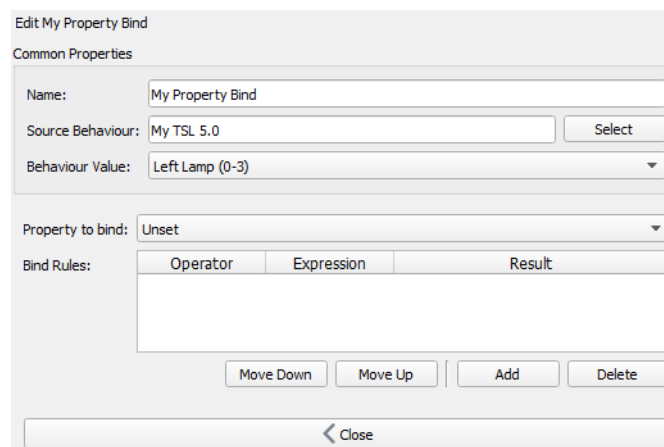


Figure 318 Property Binding Properties Dialog.

21. Change the **Name** to 'My Property Bind'.
22. Click **Select** at **Source Behaviour** and select the 'My TSL 5.0' **Behaviour**.
23. At **Behaviour Value**, 'Left Lamp (0 - 3)' should already be selected. Keep this.

This has set up which **Behaviour** data value the **Binding** will use.

Now set up the widget property to control and the rules on how to control it:

24. At **Property to Bind**, select 'Border Color' (Border > Border Color).

25. At **Bind Rules**, click **Add** four times.

Four new, default, blank **Bind Rule** rows are added to the **Bind Rules** table. See Figure 319.

| Operator | Expression | Result |
|----------|------------|--------|
| Default | | %1 |
| Default | | %1 |
| Default | | %1 |
| Default | | %1 |

Figure 319 Bind Rule Added

Note: A default **Bind Rule** simply passes on the **Behaviour** data value that the **Binding** is using (and, in some cases, this is all that is required). The Border Color needs a color value result from our **Bind Rules**.

26. In each **Bind Rule** table row, set up:
an Operator, an Expression and a Result.
See Figure 320 for the final **Bind Rules** table.

| Operator | Expression | Result |
|----------|------------|---------|
| Equals | 0 | #808080 |
| Equals | 1 | #FF0000 |
| Equals | 2 | #00FF00 |
| Equals | 3 | #FFFF00 |

Figure 320 Added Four Bind Rules to Control Border Color

27. Click **Close**.
28. Click the **Update Layout** icon.
The updated block layout is shown, see Figure 321.

Behaviour block's color indicates that the **Behaviour** used is present at 'tile-level'

Update Layout icon

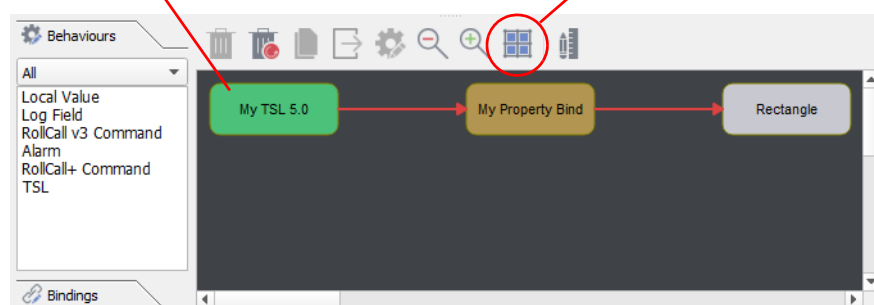


Figure 321 Rectangle Widget, Border Property Binding

29. Click the **Save** icon in the **Main Menu** to save changes to the **Tile**.

The **Rectangle** widget's border color has been configured.
Now proceed to configure the **Label** widget.

9.4.3 Label Widget

30. Click on the **Label** widget in the **Tile Editor** to show the **Behaviour and Bindings Editor** for the selected widget.

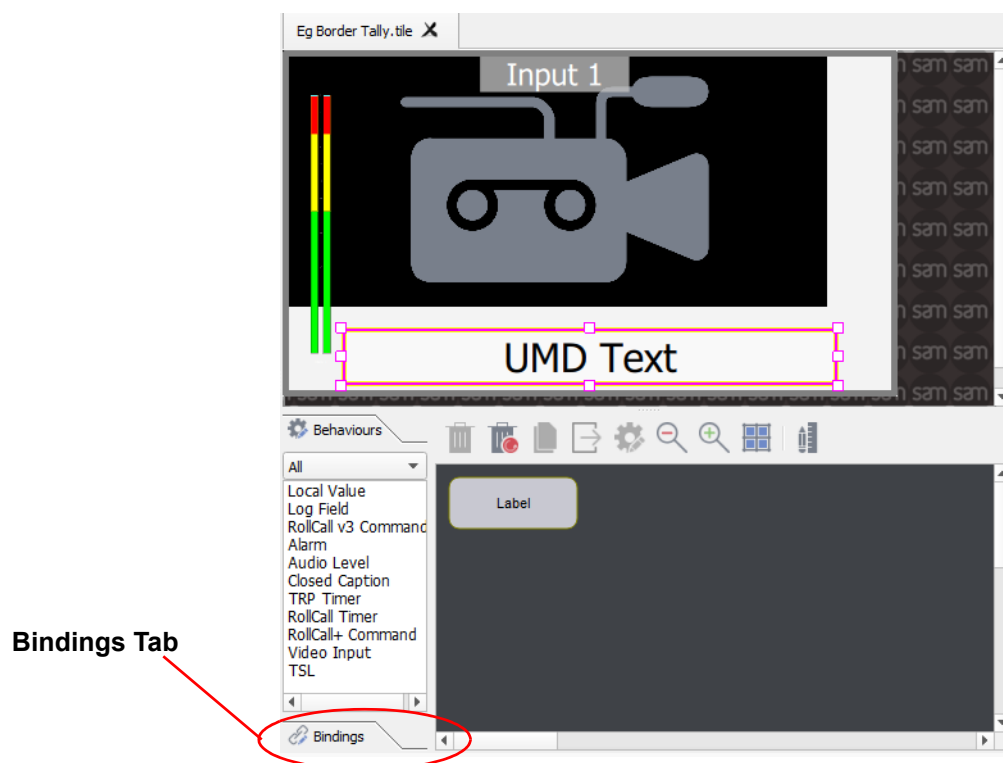


Figure 322 Behaviour and Bindings Editor for the Label Widget

31. If the list of **Behaviours** is being shown, click on the **Bindings Tab** to show the list of **Bindings**, as shown in Figure 323.

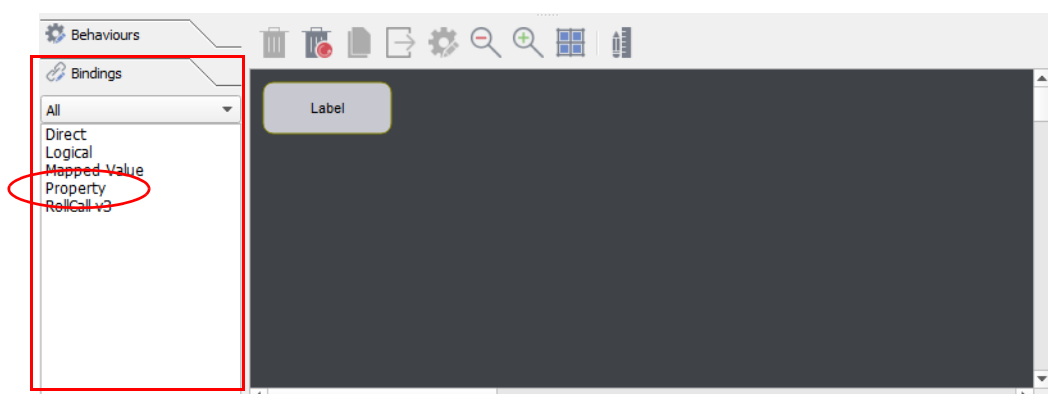


Figure 323 Showing Bindings List

32. Click three times on the **Property** name in the **Bindings** list. Three **Property Binding** blocks are added to the editor screen. See Figure 324.

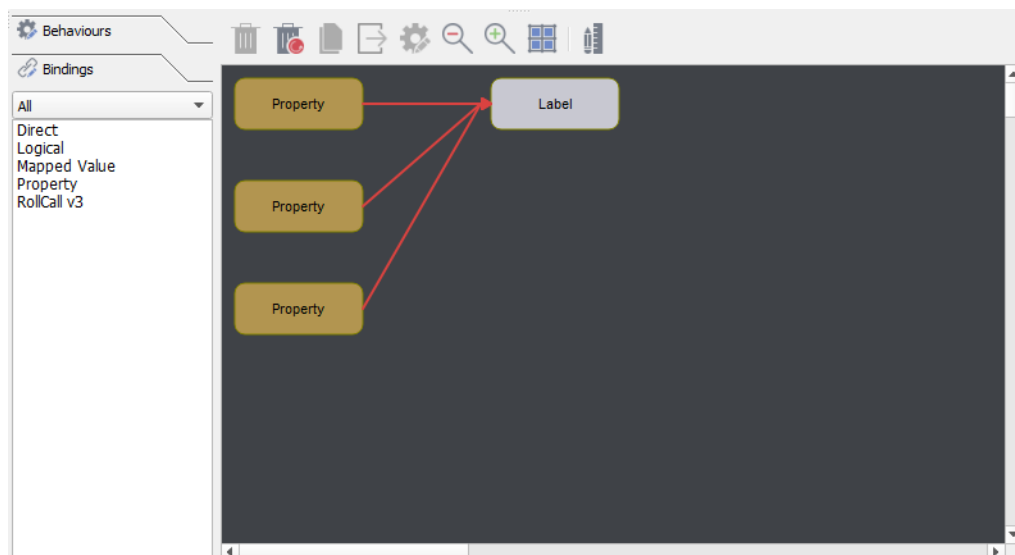


Figure 324 Three New Binding Blocks

These **Binding** blocks will be used, respectively, to control the **Label** widget's:

- 1) Caption text.
- 2) Caption text color.
- 3) Visibility.

9.4.3.1 Label Widget - Caption Text

Proceed in a similar way to Section 9.2.5 "Label Widget - Caption Text" on page 251:

33. Double-click on the first **Binding** block, to open its **Properties** dialog.
34. Set **Name** to 'My Caption Bind'.
35. At **Source Behaviour**, click **Select** and select **Behaviour** 'My TSL 5.0'.
36. At **Behaviour Value**, select **Behaviour** data value 'Label'.
37. At **Property to Bind**, select 'General > Caption'.
38. At **Bind Rules**, click **Add** once.
One empty **Bind Rule** row is added to the **Bind Rules** table.

The resulting **Bind Rule** screen is shown in Figure 325.

| Operator | Expression | Result |
|----------|------------|--------|
| | | |

Figure 325 Label Widget - Caption Binding Properties

39. Click **Close**.
40. Click the **Save** icon in the **Main Menu** to save changes to the **Tile**.

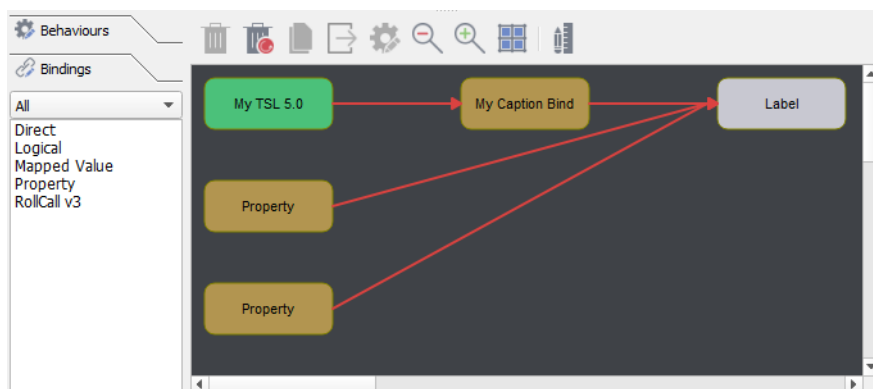


Figure 326 Label Widget, Caption Binding

The Label widget's Caption text has been configured.

9.4.3.2 Label Widget - Caption Font Color

Proceed in a similar way to Section 9.2.4 "Label Widget - Font Color" on page 249:

41. Double-click on the second **Binding** block to open its **Properties** dialog.
42. Set **Name** to 'My Font Color Bind'.
43. At **Source Behaviour**, click **Select** and select **Behaviour** 'My TSL 5.0'.
44. At **Behaviour Value**, select **Behaviour** data value 'Text Tally (0-3)'.
45. At **Property to Bind**, select 'Font > Color'.
46. At **Bind Rules**, click **Add** four times.
Four empty **Bind Rule** rows are added to the **Bind Rules** table. See Figure 327.

| Operator | Expression | Result |
|----------|------------|--------|
| Default | | %1 |
| Default | | %1 |
| Default | | %1 |
| Default | | %1 |

Figure 327 Setting up Font Color Binding

47. In each **Bind Rule** table row, set up:
an Operator, an Expression and a Result.
See Figure 328 for the final **Bind Rules** table.

| Bind Rules: | Operator | Expression | Result |
|-------------|----------|------------|-----------------------|
| | Equals | 0 | #000000 — Black |
| | Equals | 1 | #600000 — Dark red |
| | Equals | 2 | #006000 — Dark green |
| | Equals | 3 | #606000 — Dark yellow |

Move Down Move Up Add Delete

Figure 328 Added Four Bind Rules

48. Click **Close**.
49. Click the **Update Layout** icon.
The updated block layout is shown, see Figure 329.

Behaviour block color indicated that the **Behaviour** used is present at 'tile-level'

Update Layout icon

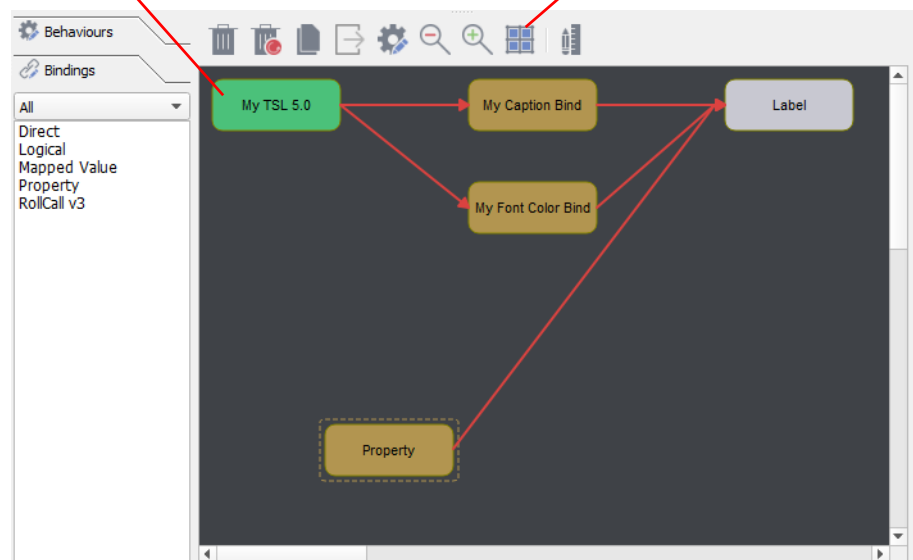


Figure 329 Label Widget, Caption and Font Color Binding

50. Click the **Save** icon in the **Main Menu** to save changes to the **Tile**.

The **Label** widget's font color has been configured.

9.4.3.3 Label Widget - Visibility

51. Double-click on the third **Binding** block to open its **Properties** dialog.
52. Set **Name** to 'My Visibility Bind'.
53. At **Source Behaviour**, click **Select** and select **Behaviour** 'My TSL 5.0'.
54. At **Behaviour Value**, select **Behaviour** data value 'Text Tally (0-3)'.
55. At **Property to Bind**, select 'General > Visibility'.

Now set up the **Bind Rules** to set the **Label**'s visibility, depending on the 'Text Tally' value:

56. At **Bind Rules**, click **Add** twice.
Two empty **Bind Rule** rows are added to the **Bind Rules** table.

57. In each **Bind Rule** table row, set up an Operator, an Expression and a Result.
See Figure 330 for the final **Bind Rules** table.

Edit Property

Common Properties

Name: My Visibility Bind

Source Behaviour: My TSL 5.0 Select

Behaviour Value: Text Tally (0-3)

Property to bind: Visibility

Bind Rules:

| Operator | Expression | Result |
|--------------|------------|--------|
| Greater Than | 0 | True |
| Default | 0 | False |

Move Down Move Up Add Delete

Figure 330 Setting up Visibility Binding

58. Click **Close**.
59. Click the **Update Layout** icon.
The updated block layout is shown, see Figure 331.

Behaviour block color indicates that the **Behaviour** used is present at 'tile-level', or on another widget

Update Layout icon

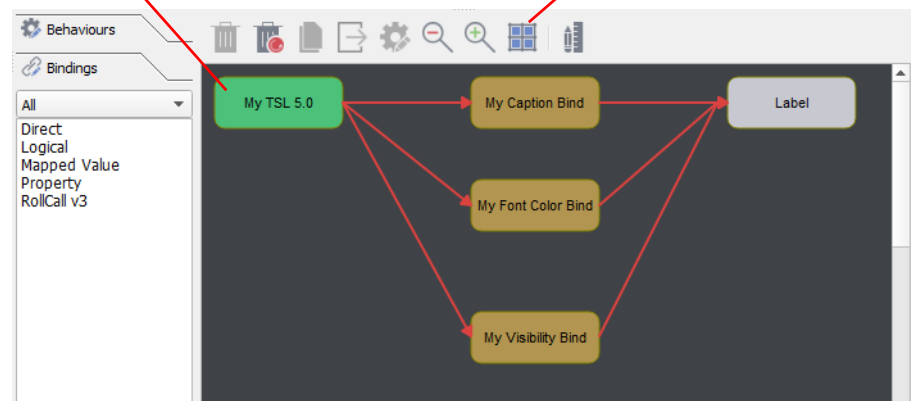


Figure 331 Label Widget, Caption and Font Color Binding

60. Click the **Save** icon in the **Main Menu** to save changes to the **Tile**.
The **Label** widget's visibility has been configured.

9.4.3.4 Finishing the Layout

The **Behaviours** and **Bindings** have been configured. Now, finally:

61. Resize and reposition the **Video** widget, so that it within the **Rectangle** widget border.
(See Figure 332 for the **X** and **Y Position**, and **Width** and **Height Size** property values.)
62. Click the **Save** icon in the **Main Menu** to save changes to the **Tile**.

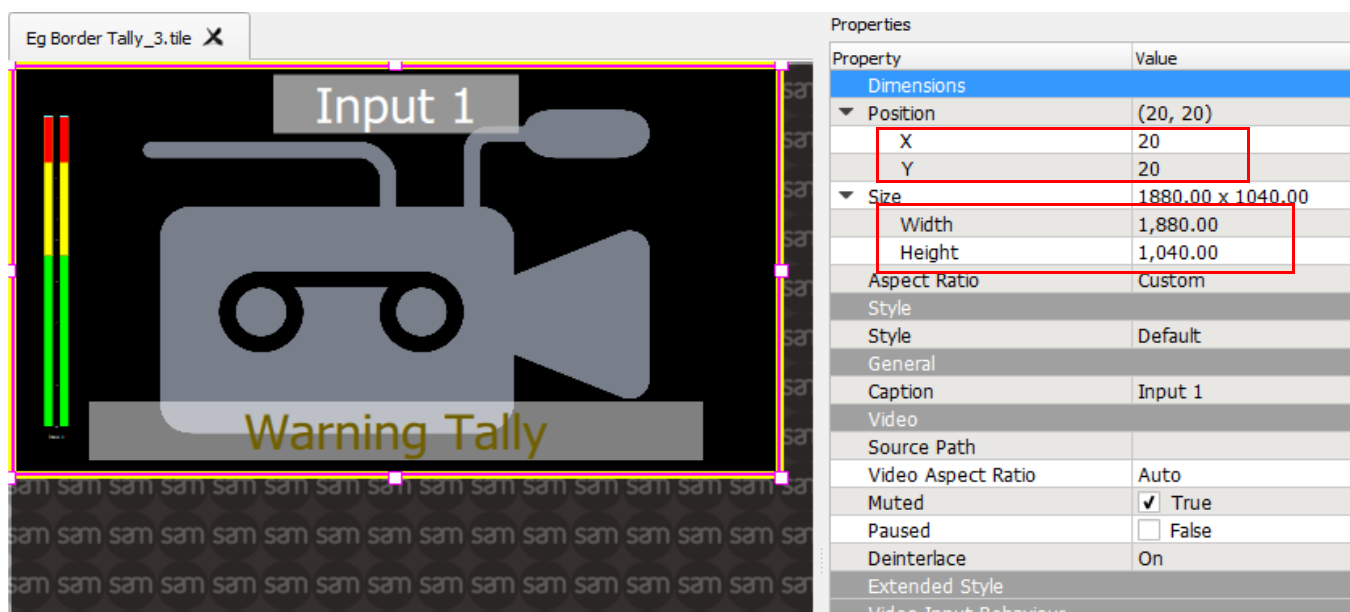


Figure 332 Completed Border Tally

9.4.4 Border Tally Behaviour and Bindings Recap

The **TSL Behaviour** (a TSL Listener) gets data from a TSL server in the system. The **TSL Behaviour** presents information about a certain video feed to the **Tile** widgets. Figure 333 summarizes the **Behaviour** and **Bindings** set up.

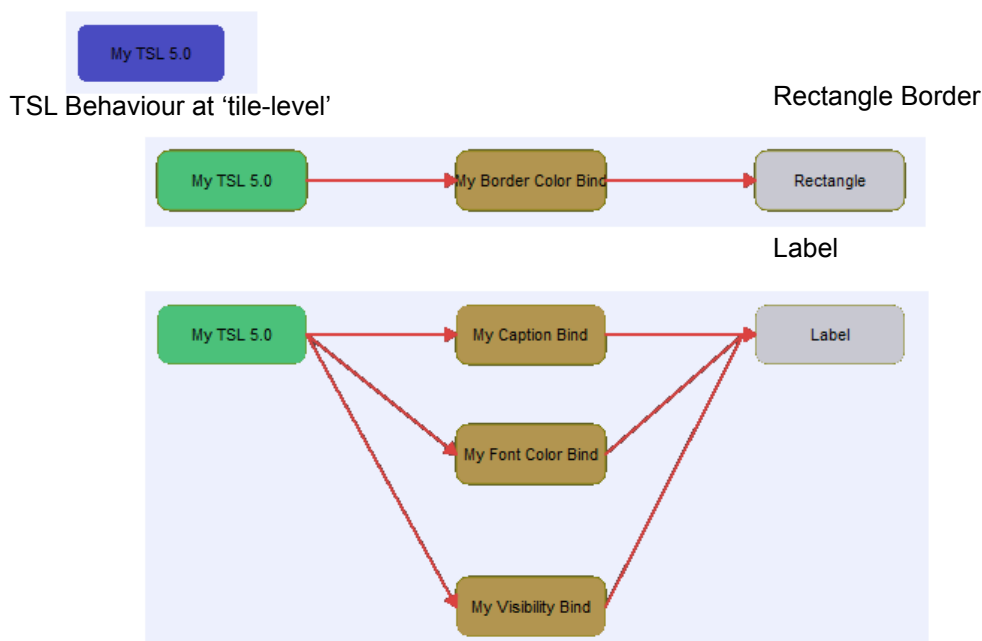


Figure 333 Border Tally Behaviours and Bindings

Figure 334 shows the on-screen appearances. In this example, the border is always present and the video lies fully within the border.

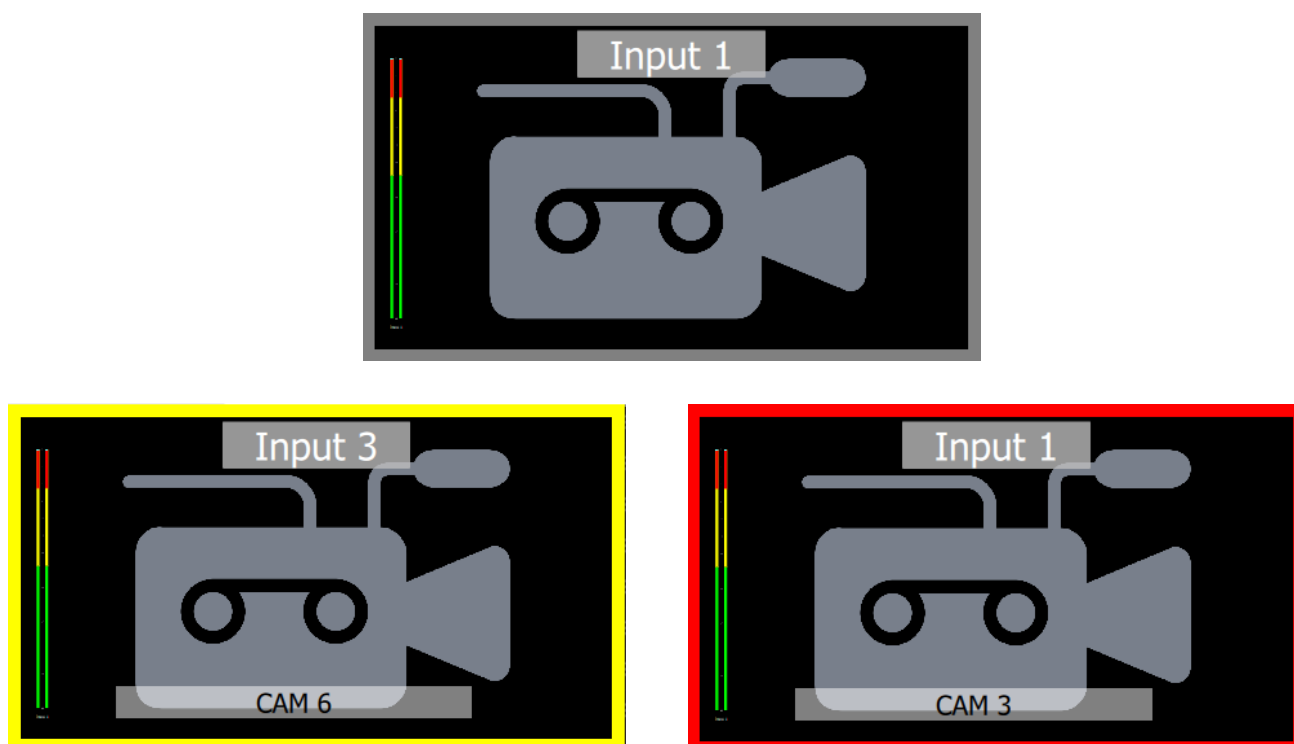


Figure 334 Border Tally Examples

An alternative Border Tally design can show full-sized video and only show a border when a Tally is asserted. See 9.5 “Alternative Border Tally” on page 280.

9.5 Alternative Border Tally

Section 9.4.4 “Border Tally Behaviour and Bindings Recap” describes a Border Tally example which has an always-visible rectangular border and video within the border.

An alternative Border Tally design may show video on the full-size of the tile, with no border, and then only show a border when a tally is asserted.

In this case, the border's visibility (i.e. the Rectangle widget's visibility) can be controlled by TSL Tally data. This allows the border to be normally off and only visible when a tally is asserted, overlaying the video.

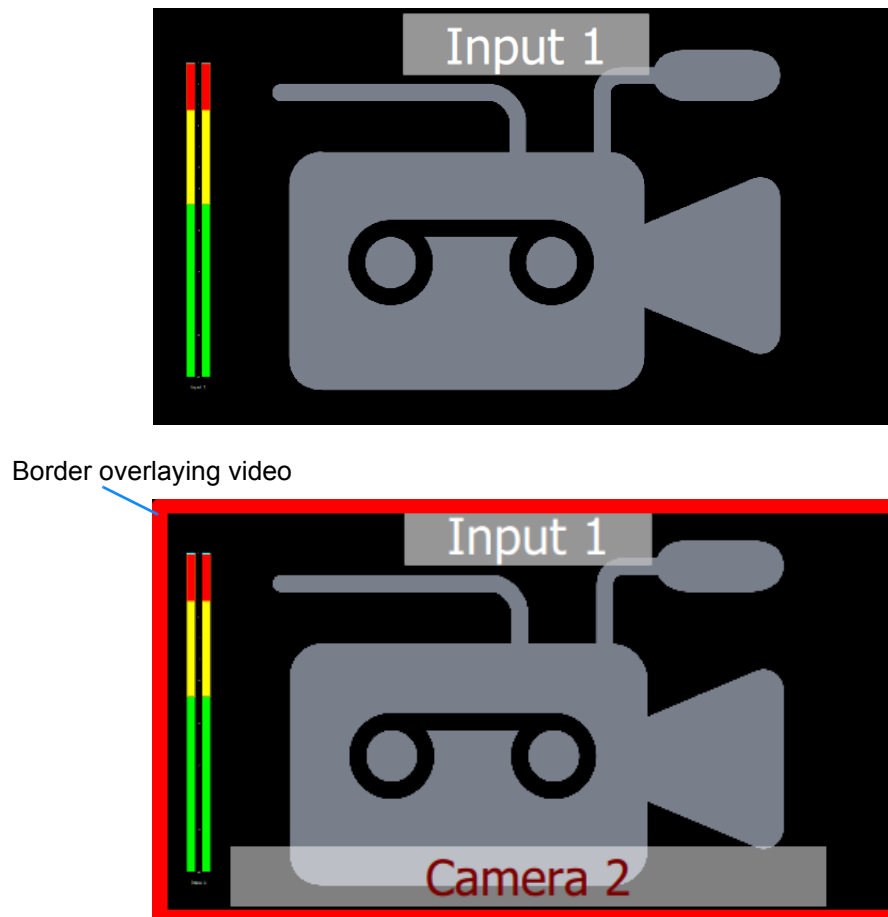


Figure 335 Alternate Border Tally - Border and Text Label Change Visibility

Start from the Border Tally example above in Section 9.4.4 and change it to have the behavior of our Alternate Border Tally design.:

1. Open the Border Tally example.
2. Select the **Rectangle** widget and click the **Edit Behaviours** icon.
The **Behaviours and Bindings Editor** for the selected widget is shown.
3. Add a **Property Binding**, a new **Binding** block appears in the editor.
4. Double-click on the new **Binding** block.
The **Binding** properties dialog is shown.
5. Change **Name** to 'My Visibility Bind'.
6. Select 'My TSL 5.0' as **Source Behaviour**.
7. Keep **Behaviour Value** as 'Left Lamp (0-3)'.
8. At **Property to Bind**, select 'General > Visibility'.

Now set up the **Bind Rules** to set the **Rectangle**'s visibility, depending on the 'Left Lamp' value:

9. At **Bind Rules**, click **Add** twice.
Two empty **Bind Rule** rows are added to the **Bind Rules** table.
10. In each **Bind Rule** table row, set up:
an Operator, an Expression and a Result.
See Figure 336 for the final **Bind Rules** table.

Figure 336 shows the 'Edit My Visibility Bind' dialog box. The 'Common Properties' section includes:

- Name: My Visibility Bind
- Source Behaviour: My TSL 5.0
- Behaviour Value: Left Lamp (0-3)
- Property to bind: Unset

The 'Bind Rules' section contains a table with the following data:

| Operator | Expression | Result |
|--------------|------------|--------|
| Greater Than | 0 | True |
| Default | | False |

Buttons at the bottom include Move Down, Move Up, Add, and Delete.

Figure 336 'Binding for Rectangle Visibility

11. Click **Close**.
12. Click the **Update Layout** icon.
The updated block layout is shown, see Figure 337.

Behaviour block color indicated that the **Behaviour** used is present at 'tile-level'

Update Layout icon

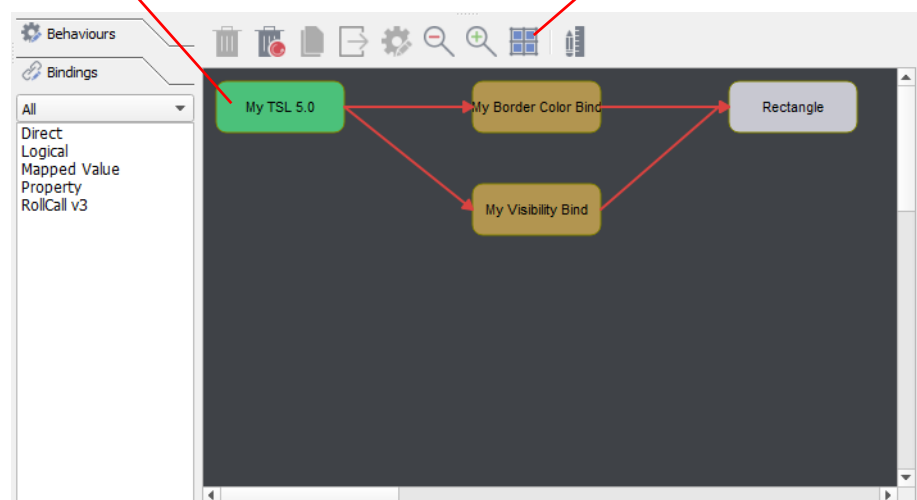


Figure 337 Rectangle Widget, Visibility Binding

13. Click the **Save** icon in the **Main Menu** to save changes to the **Tile**.

The **Rectangle** widget's visibility has been configured.

Finally:

14. Resize and reposition the **Video** widget to fill the tile area.
See Figure 338 for the **X** and **Y Position**, and **Width** and **Height Size** property values for our example.

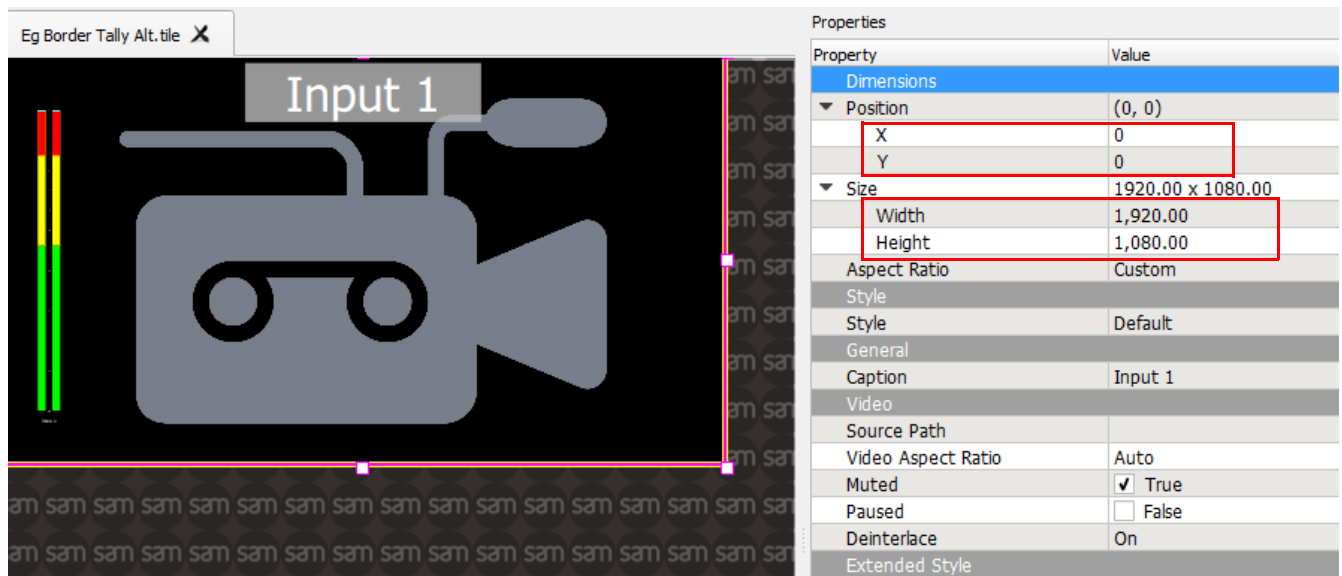


Figure 338 Completed Border Tally

15. Click the **Save** icon in the **Main Menu** to save changes to the **Tile**.

9.6 Exercise the Tile and Widgets

To exercise the widgets on a **Tile/Schematic**:

1. Select the widget to be shown in the Behaviours and Bindings Editor.
2. Click the Behaviours and Bindings **Run Mode** icon in the Behaviour and Binding editor. See Figure 339.

The tile and the widgets it contains are exercised.

The selected widget is shown in the Behaviours and Bindings Editor, see Figure 339.

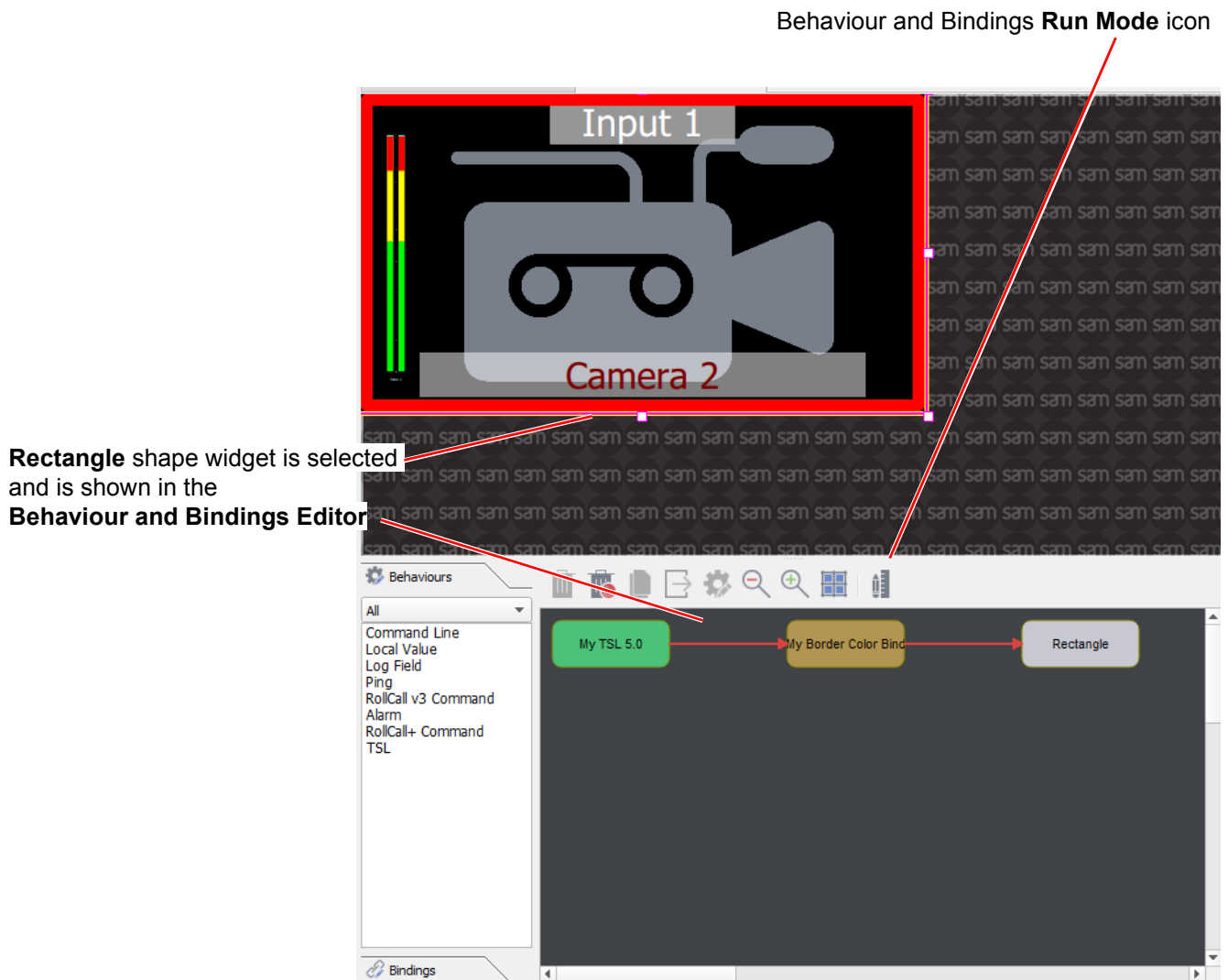


Figure 339 Exercise the Alternative Border Tally Example

When the tile is in **Run Mode**, all variables will initially take default values and then values of Behaviour and modified properties are shown. See Figure 340.

Figure 340 shows two on-screen views of the tile being exercised as it reacts to different Behaviour values. The tile's border **Behaviour** and property value are shown in the **Behaviour and Bindings Editor** (Figure 340a and Figure 340b).

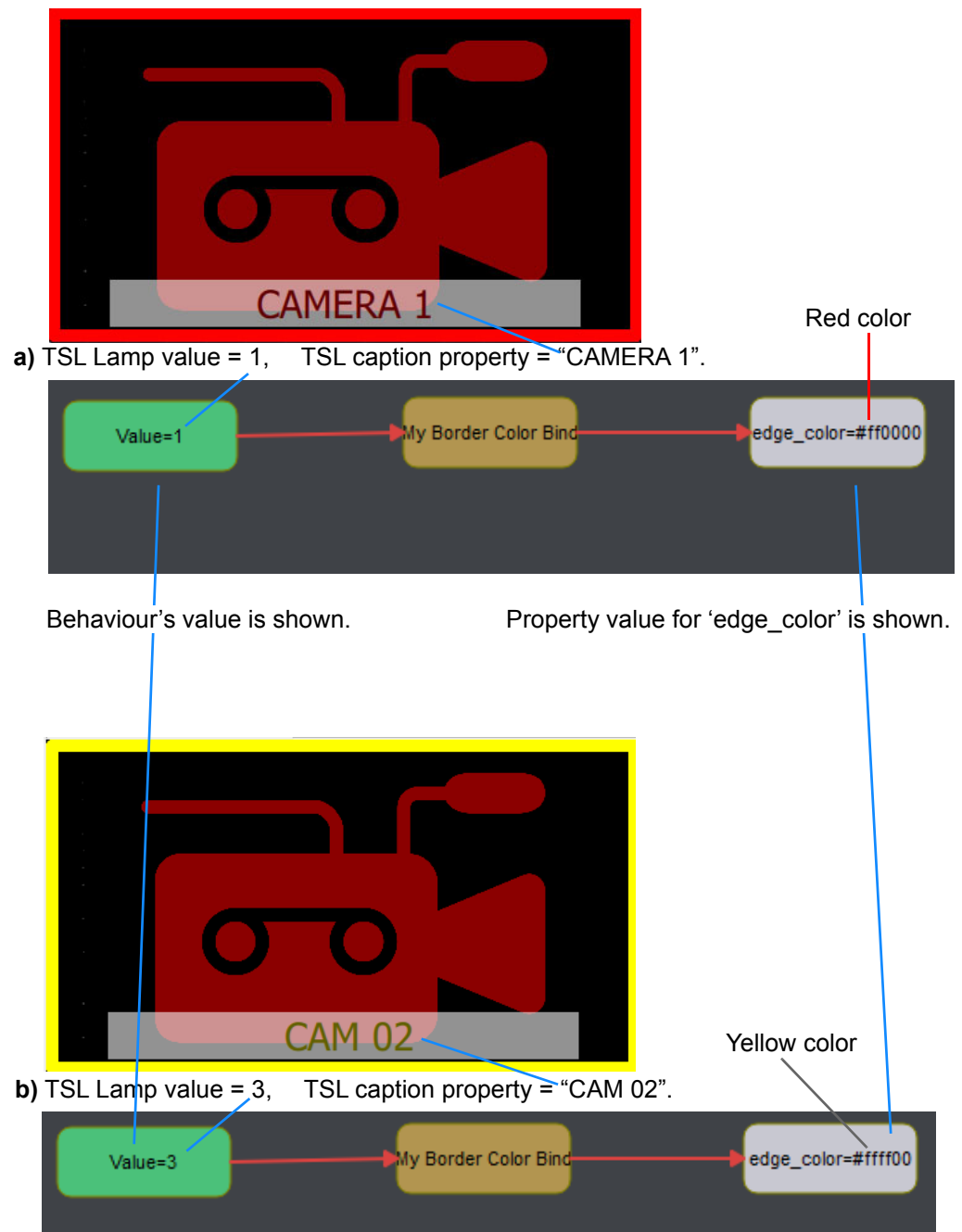


Figure 340 Exercising the Alternative Border Tally Example:

- a) TSL Lamp Behaviour value = 1.
- b) TSL Lamp Behaviour value = 3.

3. To revert to **Design Mode** (for making changes to Tiles, Widgets, Behaviours and Bindings), click the **Run Mode** icon again.

Note:

For a UMD widget or for the Border Tally examples here which use TSL data, there needs to be live TSL data in the system to be able to exercise this on screen.

For example TSL source data from a video switcher or a control system.

I.e. if Orbit is connected to a network where TSL messages are being sent to a multiviewer, then a UMD tile will show information from the various TSL tallies for the default TSL addresses set up in widget properties.

10

10 Examples - Other On-screen Functionality

Examples - Other On-screen Functionality

| | |
|--|-----|
| On-screen Notification from RollCall Log Server | 286 |
| Preliminaries | 287 |
| Setting up the Lower Label - On-Screen | 288 |
| Setting Up the Upper Label - Behaviour | 289 |
| Upper Label - Binding Background Fill Color | 291 |
| Image Widget Set Up | 293 |
| Finally | 295 |
| Adding a GPI Output to a Tile | 297 |
| Setting up a Multiviewer's GPIO to be an Output | 298 |
| Custom Tile and GPIO | 299 |
| Auto-Tile and GPI Output | 303 |
| On-Screen Video Input Status | 306 |
| Create a Video Tile Style Showing Text Lines | 306 |
| Set Up the Number of On-Screen Text Status Lines | 308 |

This section contains some example uses of Orbit's Behaviours and Bindings to carry out some common functions.

Note:

Selection/ Deselection of graphical items when editing items in the Tile Editor:

| Select / Deselect | Action |
|------------------------------|---|
| Deselect All | Click outside the schematic area. Or press the ' Esc ' keyboard key. |
| Select | Click on an item to select it. Alt-Click selects another item: <ul style="list-style-type: none"> • In a stack of items. • Or selects the Group Layout that an item belongs to. |
| Add Select | Shift-Click on an item to add it to current selection. Shift-Alt-Click to add another item from a stack of items to the current selection. |
| Remove from Selection | Shift-Click on an <i>already-selected</i> item to remove it from the current selection. |
| Select All | ' Ctrl-A ' selects all items. |
| Select an Area | Shift drag cursor on screen to describe a rectangular area. All objects wholly or partly in the area will be selected. |

10.1 On-screen Notification from RollCall Log Server

This example explains how to create a **Tile** linked to a RollCall Log Server **Behaviour** for use in on-screen notifications.



Figure 341 Creating a System Variable

An advanced Orbit multiviewer video wall or a Orbit control and monitoring screen may be designed to effectively be a Grass Valley MapView screen with high resolution video. In these advanced cases, a common requirement is for various system names and statuses to be reported on-screen.

If a system has a Grass Valley RollCall Log Server, then any logged system information can be shown on the Orbit screen or multiviewer video wall.

This example creates a status notification tile, see Figure 342, where the tile's widgets are connected to some system status information obtained from a RollCall Log Server.

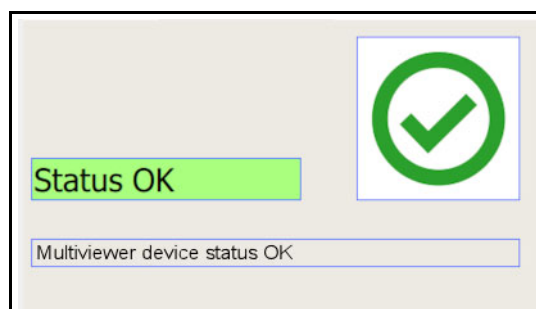


Figure 342 Example Status Notification Tile

The tile will use a **RollCall Behaviour**.

The lower text will be set up to show the Log Message's text.

The upper text will be set up to show a simpler on-screen message in a color-coded way.

10.1.1 Preliminaries

RollCall Logserver settings in the Orbit project:

1. In the **Multiviewer > Properties** menu, set up the RollCall Log Server domain number (**LogServer Domain**). and RollCall Log Server's IP address (**LogServer IP 1**).

This will enable a connection with the Log Server for exercising the **Tile**.

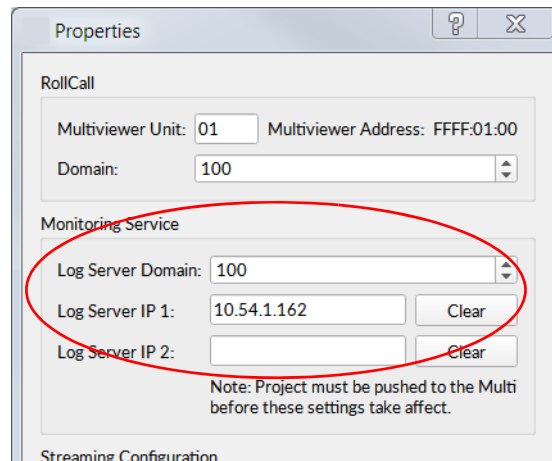


Figure 343 Multiviewer > Properties Menu - RollCall Log Server Information

Importing graphics:

The widget uses some user-defined images which must be imported into the Orbit project. See Figure 344.



a) icon_ok.png



b) icon_warning.png



c) icon_error.png

Figure 344 Images for Import

To import image files into the Orbit project:

2. Expand the **Project View** to the 'resources > images' sub-folder. Right-click on 'images' and click **Import Files**.
3. Browse to each image file to be imported and click **Open**.

After importing, the file names are shown under 'resource > images'.

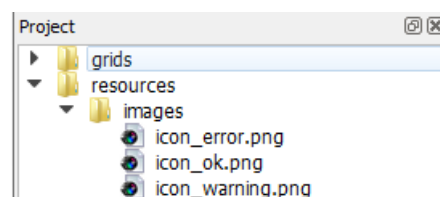
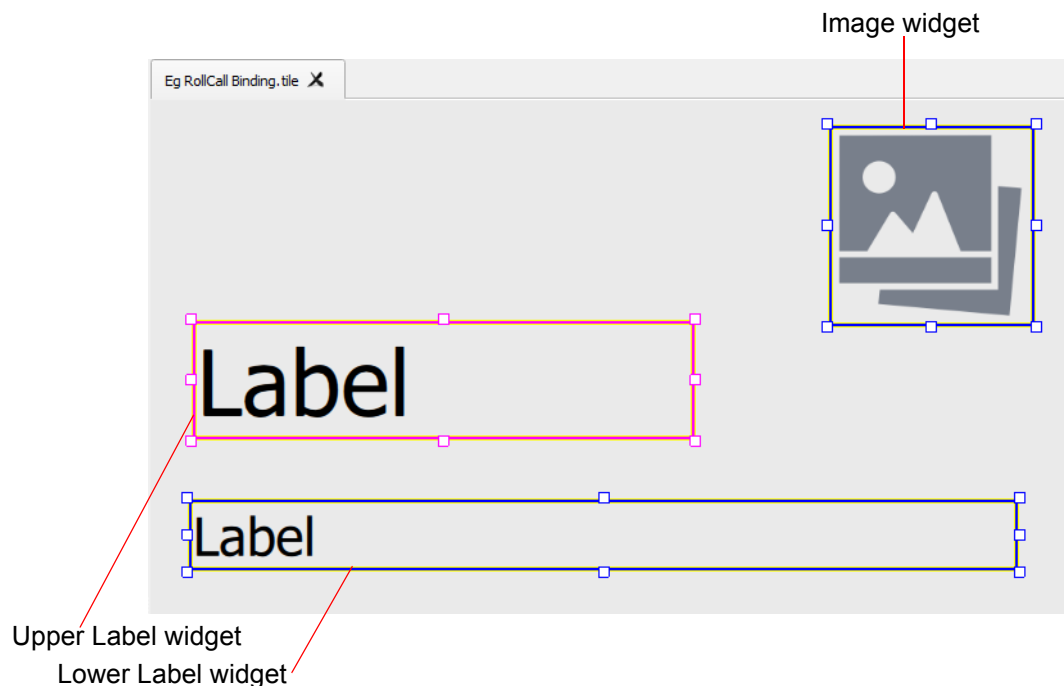


Figure 345 Imported Images

4. Click **File > Save Project**.

Creating the Tile:

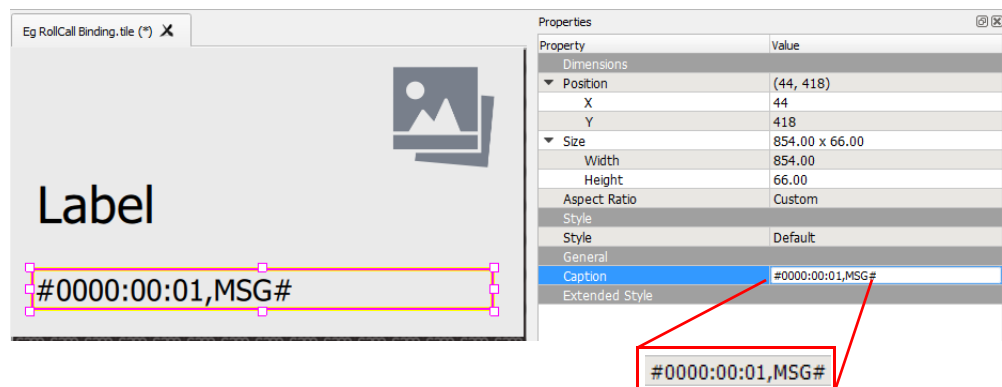
5. Create a new blank **Tile** and open it in the **Tile Editor**.
6. Add two **Label** widgets and one **Image** widget to the tile. Resize the widgets to suit, see Figure 346.

**Figure 346 Adding Widgets to a Tile**

The basic tile has been created. Each widget now needs to have its functionality set up.

10.1.2 Setting up the Lower Label - On-Screen

7. Click on the lower **Label** widget to select it. The widget properties are shown in the **Property Box**.
8. In the **Property Box**, change the **Caption** property to the value shown in Figure 347 - where "0000:00:01" is the RollCall address of the device whose status is to be monitored. Replace "0000:00:01" with the RollCall address of your device of interest.

**Figure 347 Caption Text to Show a RollCall status MSG Message Content**

9. Click **Save File** icon and click **File > Save Project**.

The lower text is now set up.

Note:**Special Shorthand:**

The text string used here, “#0000:00:01,MSG#”, is a special Orbit text shorthand which configures the label caption to show the contents of a RollCall “MSG” type message from RollCall Address 0000:00:01.

10.1.3 Setting Up the Upper Label - Behaviour

The upper Label widget will monitor status messages and display abbreviated on-screen status. This requires some **Behaviours** and **Bindings**.

Create a Behaviour to Monitor Log Messages:

10. Click on the **Edit Behaviours** icon to show the **Behaviours and Bindings Editor**.
11. Click on the upper **Label** widget to select it.
12. Click on ‘Log Field’ in the **Behaviours** tab.

A **Log Field Behaviour** block is added to the editor screen. See Figure 348.

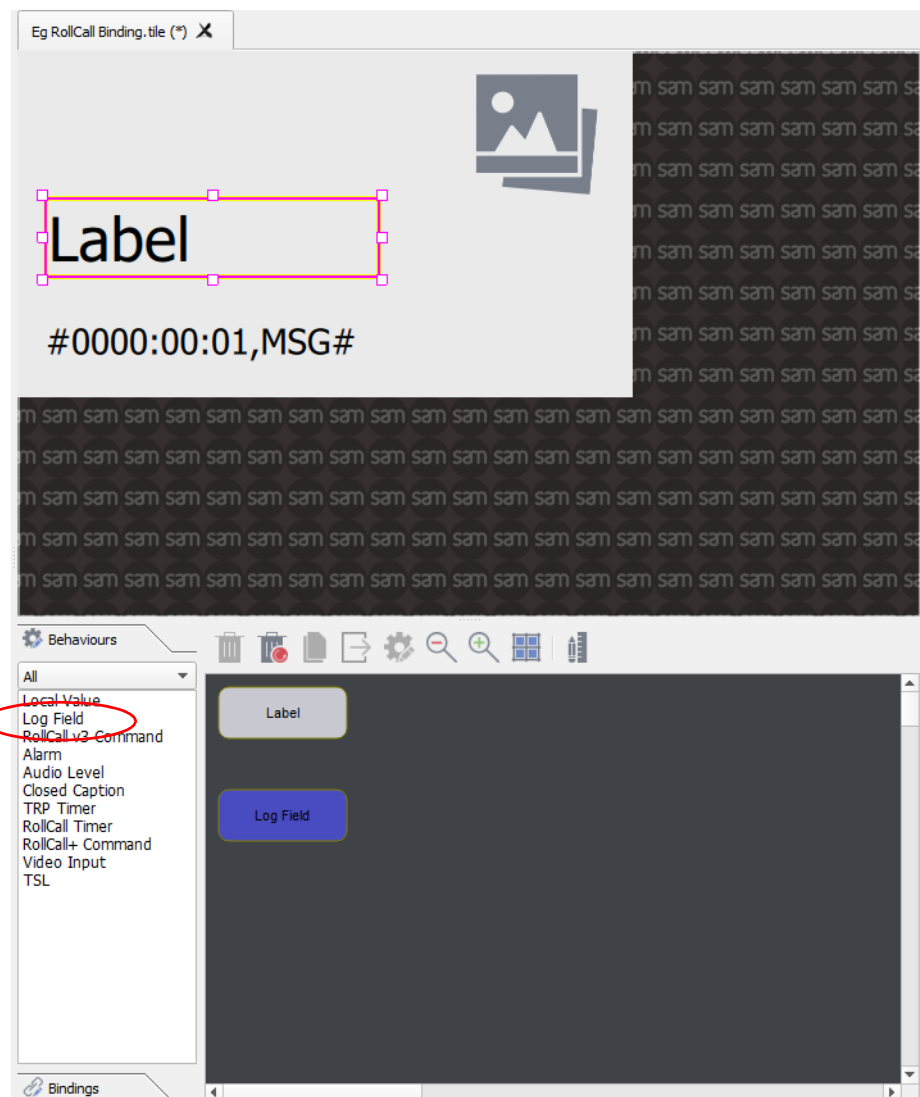



Figure 348 Log Field Behaviour

13. Double-click on the **Log Field Behaviour** block.
The **Behaviour** properties dialog is shown.

14. Change **Name** to 'My Log Field Beh'.
15. Change **RollCall Address** to be the RollCall address of the unit to be monitored.
16. Leave the **Number of Headers** value set to '1'.
17. Click in the **Headers** *value* field
and click on the  icon on the right, see Figure 349.

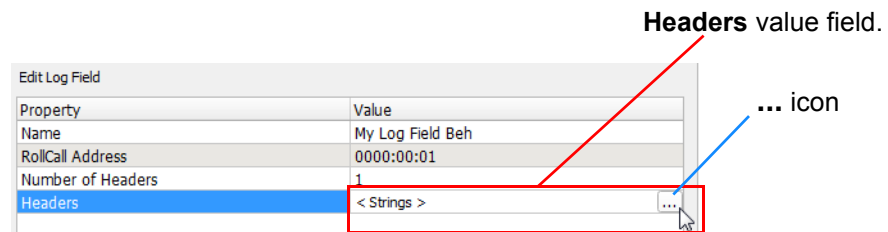


Figure 349 Log Field Behaviour Properties

The **Strings Editor** dialog is then displayed, see Figure 350a.

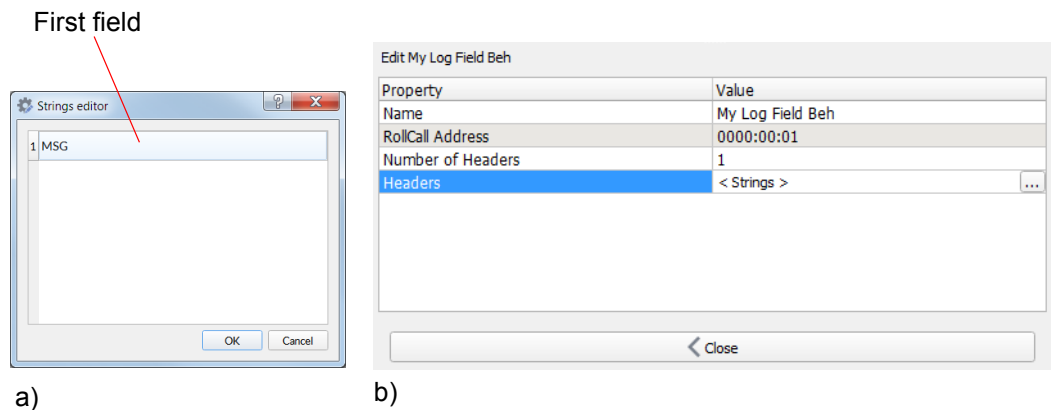


Figure 350 a) Strings Editor; b) Completed Behaviours Properties dialog.

18. In the **Strings Editor**, click in the first field
and enter "MSG", see Figure 350a.
19. Click **OK**.
The completed **Behaviours** properties dialog is shown in Figure 350b.

Note:

"MSG" Keyword:

The text "MSG" in the **Headers** string occurs in the RollCall log message header and it states the overall status of a device. "MSG" acts as a keyword in the required log message heading.

20. Click **Close**.
21. Click the **Save File** icon.

This has created a **Behaviour** named 'My Log Field Beh' which monitors "MSG" log messages from the unit to be monitored.

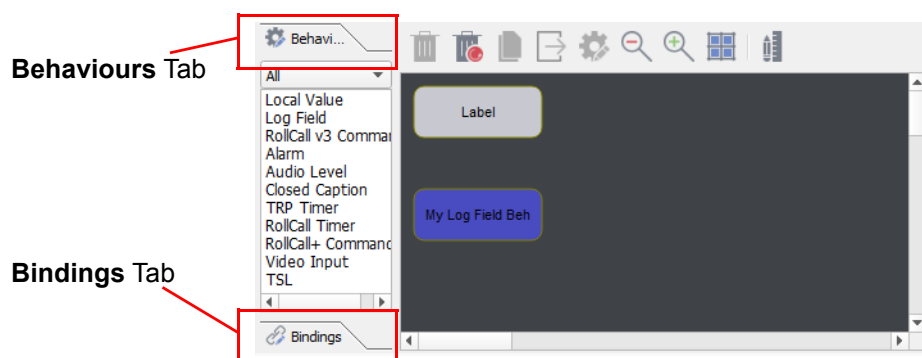


Figure 351 Log Field Behaviour Block

10.1.4 Upper Label - Binding Background Fill Color

22. Select the upper **Label** widget.
23. Click on the **Bindings** tab and click on the **Properties** name in the **Bindings** tab. A **Properties Binding** block is added to the editor screen.
24. Double-click on the **Properties Binding** block. The **Binding** properties dialog is shown, see Figure 352.

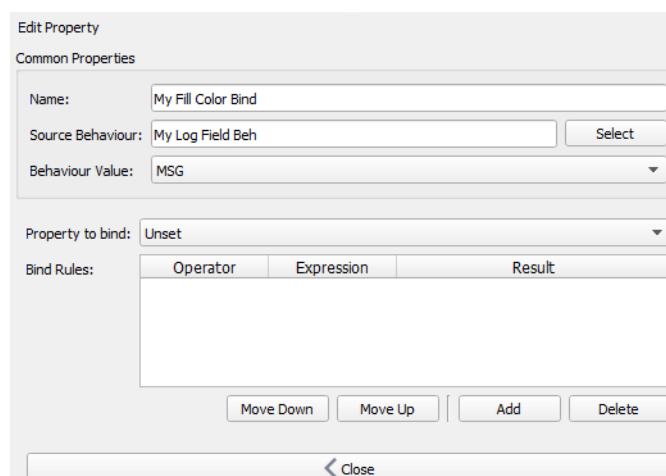


Figure 352 Binding Properties Dialog

25. Change **Name** to 'My Fill Color Bind'.
26. Select **Source Behaviour** 'My Log Field Beh'.
27. Leave **Behaviour Value** as 'MSG'.
28. Select **Property to Bind** to be 'Fill Color' (Fill > Fill Color).

29. Under **Bind Rules**, click **Add** three times.
Three new blank **Bind Rule** items appear in the box, see Figure 353.

| Operator | Expression | Result |
|----------|------------|--------|
| Default | | %1 |
| Default | | %1 |
| Default | | %1 |

Figure 353 Three Blank Bind Rules

30. For each **Bind Rule** row, edit each column item (Operator, Expression, Result).
The rules will look for log messages containing either the text “OK”, “WARN” or “FAIL” and set a color result accordingly. The result is bound to Label widget’s Fill Color property.

The resulting **Bind Rules** table is shown in Figure 354.

| Operator | Expression | Result |
|---------------|------------|--------------|
| Contains Text | OK | Light Green |
| Contains Text | WARN | Light Yellow |
| Contains Text | FAIL | Light Red |

Figure 354 Completed Bind Rules for Fill Color

31. Click **Close**.
32. Click the **Save File** icon.
33. Click the Update Layout icon.
The resulting **Behaviours and Bindings Editor** screen for the upper Label widget is shown in Figure 355.

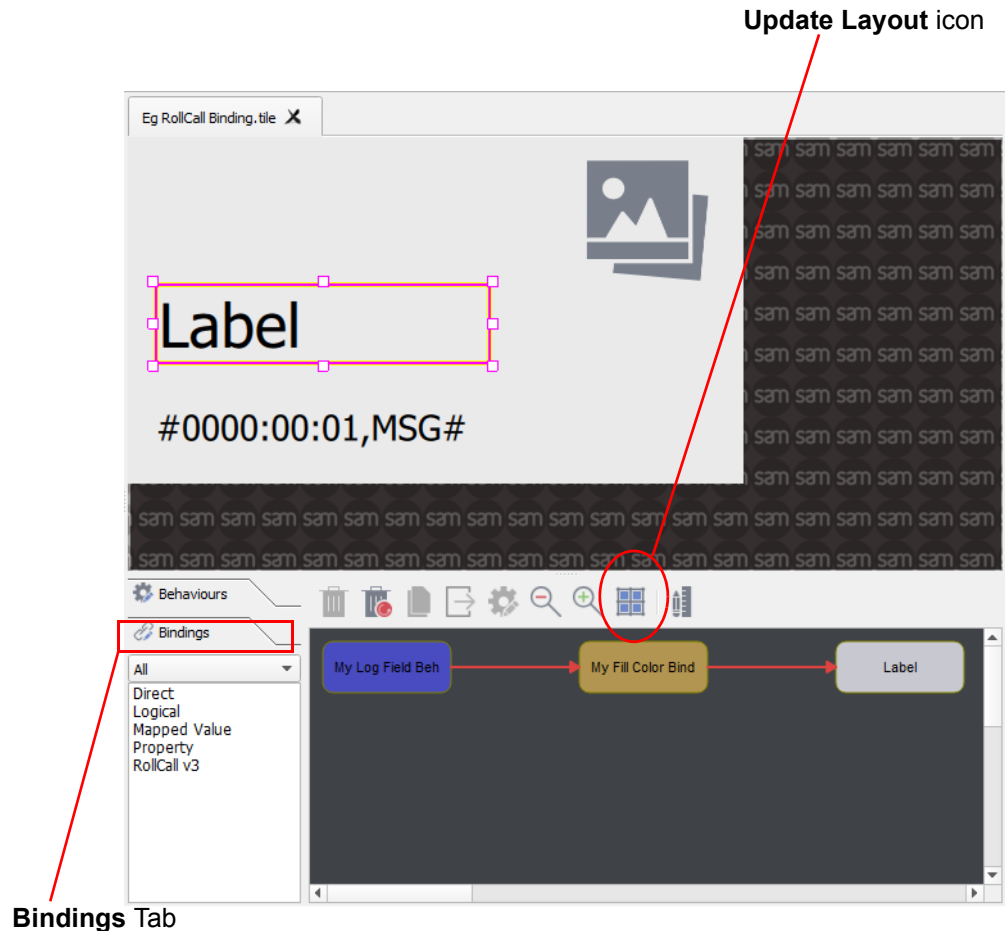


Figure 355 Upper Label Behaviours and Bindings

This has defined label fill colors for the upper **Label** widget for “OK”, “WARNING” and “FAILure” conditions reported by the log messages being monitored.

The upper text label is now set up.

10.1.5 Image Widget Set Up

We now introduce a binding for the **Image** widget, which will change the image shown according to the log message:

The **Behaviour** that will be used for the **Image** widget is the ‘My Log Field Beh’ **Behaviour** already defined for the **Tile**.

34. Select the **Image** widget on the **Tile**.
The corresponding (blank) **Behaviours and Bindings Editor** screen is shown.
35. Click on the **Property Binding** name in the **Bindings** list to add a **Property Binding** block to the editor screen.
36. Double-click on the **Property Binding** block to display the properties dialog.
37. Set **Name** to ‘My Image Bind’.
38. Select **Source Behaviour** ‘My Log Field Beh’.
39. Leave **Behaviour Value** as ‘MSG’.
40. Select **Property to Bind** to be ‘Source Path’ (Image > Source Path).

41. Under **Bind Rules**, click **Add** three times.
Three new blank **Bind Rule** items appear in the box, see Figure 356.

| Operator | Expression | Result |
|----------|------------|--------|
| Default | | %1 |
| Default | | %1 |
| Default | | %1 |

Figure 356 Three Blank Bind Rules

42. For each **Bind Rule** row, edit each column item (Operator, Expression, Result).
The rules will look for log messages containing either “OK”, “WARN” or “FAIL” and set an image path result accordingly.
The image path is the path to the image file within the Orbit project.
The completed table of **Bind Rules** should look similar to Figure 357.

| Operator | Expression | Result |
|---------------|------------|------------------------------------|
| Contains Text | OK | /resources/images/icon_ok.png |
| Contains Text | WARN | /resources/images/icon_warning.png |
| Contains Text | FAIL | /resources/images/icon_error.png |

Figure 357 Binding for Image Widget

43. Click **Close**.
44. Click the **Update Layout** icon.

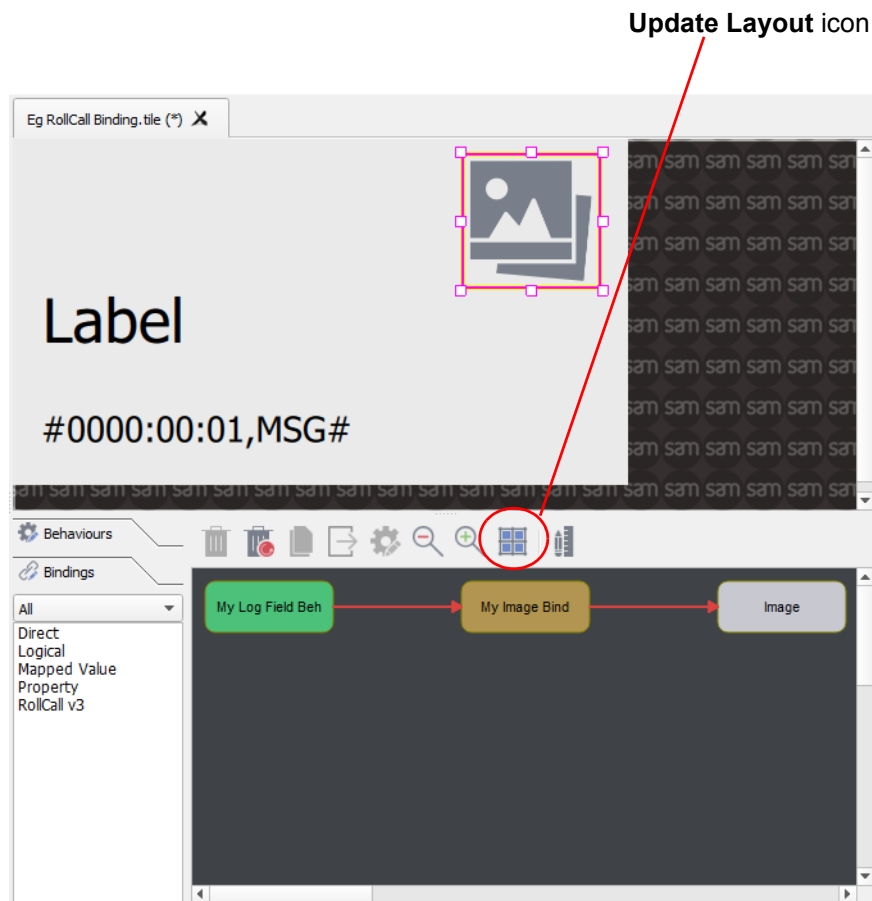


Figure 358 Image Widget Behaviours and Bindings

45. Click the **Save File** icon.
46. Click **File > Save Project**.

This has defined images to be displayed on the **Image** widget for “OK”, “WARNIng” and “FAILure” conditions reported by the log messages being monitored.

The **Image** widget is now set up.

10.1.6 Finally

47. Close the **Tile Editor**.
The tile name appears as a .tile file under the ‘tiles’ folder within the Orbit **Project View**.
The tile may be used on a multiviewer video wall or MapView screen.

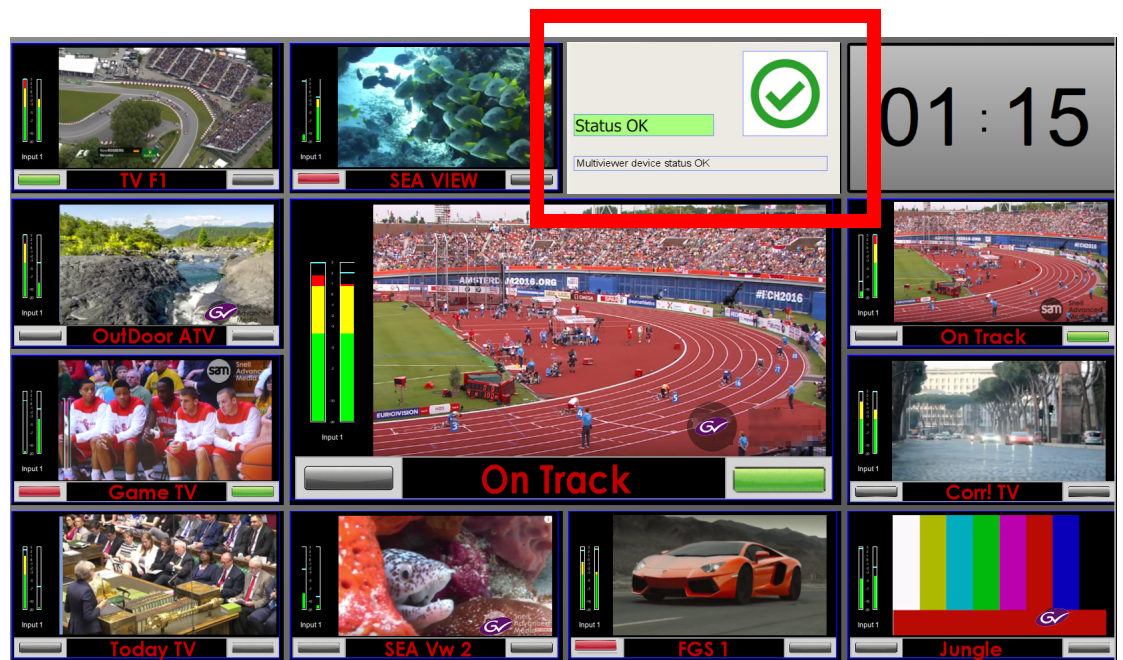


Figure 359 'RollCall Binding Tile

The tile may be used on a multiviewer video wall or MapView screen.

10.2 Adding a GPI Output to a Tile

This example explains how to add control over a multiviewer's GPI Output to a video tile, so that a tile could trigger an external alarm device.

The video wall design and configuration is done in the Orbit application by adding GPI output functionality to a Video widget in a video wall tile and configuring the GPI output of the multiviewer.



Figure 360 GPI Output from a Tile

The sub-section covers:

- Configuring a video tile; the following tile types are covered:
 - **Custom Tile** using a **RollCall+ Behaviour** and a **Mapped Value Binding**.
 - **Auto-Tile** by setting wall-level properties.
- Configuring a GPI Output.

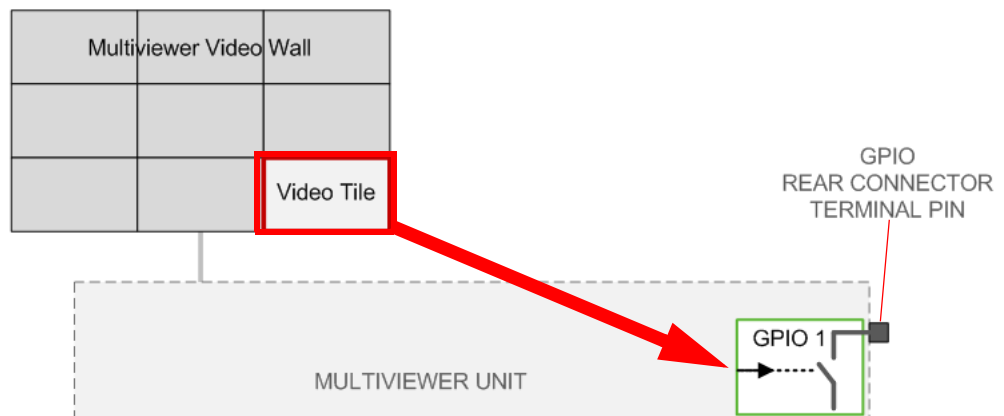


Figure 361 Video Tile and GPIO

10.2.1 Setting up a Multiviewer's GPIO to be an Output

This section configures the GPIO output of a multiviewer via the Grass Valley RollCall Control Panel. Our example deals with GPIO 1:

1. Run Grass Valley RollCall Control Panel and connect to the multiviewer unit. The multiviewer's RollCall templates are displayed.
2. Select the GPIO template screen. See Figure 362.

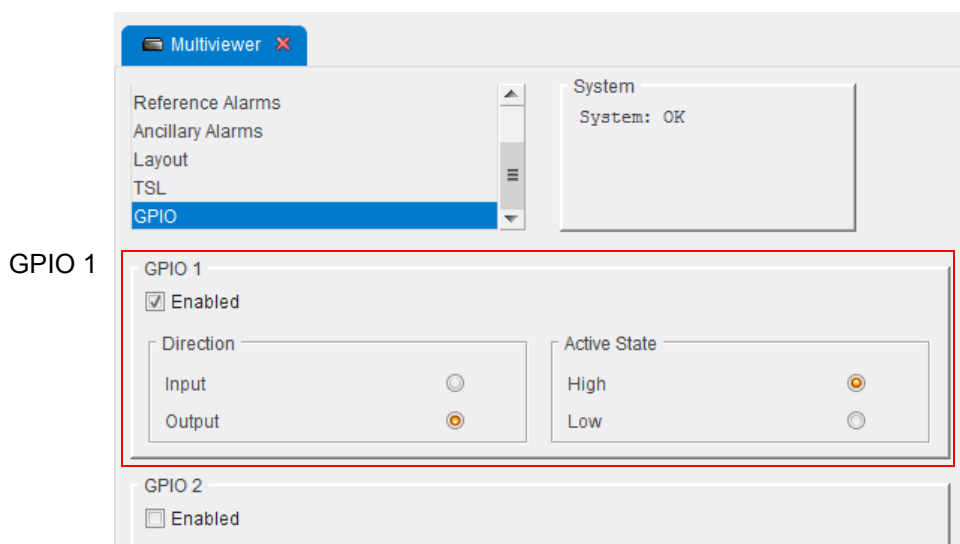


Figure 362 GPIO RollCall Template Screen

There are a number of multiviewer GPIO ports and this example uses GPIO number 1. (For GPI rear connector details, see the corresponding multiviewer User Manual.)

For GPIO 1 settings on the RollCall template:

3. Select the **Enabled** check box.
4. Set **Direction** to be 'Output'.
5. Set the **Active State** of the GPI, 'High' or 'Low' depending on the requirements of the external device connected to GPIO 1. See Table 28.

| Active State Setting | GPIO Connector pin voltage (See Note 1) | |
|----------------------|--|-----------------------------|
| | GPIO De-asserted | GPIO Asserted (i.e. Active) |
| High | Low | High |
| Low | High | Low |

Note 1: For pin voltage details, see corresponding multiviewer User Manual.

Table 28 GPIO Active State Setting

6. Close the RollCall Control Panel.

This has configured the multiviewer's GPIO 1 hardware to be an output.

10.2.2 Custom Tile and GPIO

The **Tile** used in this example is a **Custom Tile** containing a **Video** widget, which may have originally been from an **Auto-Tile**. **Behaviours** and **Bindings** are added to give the tile some GPIO Output functionality (a GPIO output will assert if there is an unacknowledged error on the **Tile**'s video input).

Initially, a **Behaviour** is added to the **Tile** to control GPIO output GPIO 1 via a RollCall command:

1. In the **Tile Editor**, edit a **Custom Tile** which contains a video widget.
2. Deselect all widgets on the tile by clicking the **Tile Editor** background.
3. Click the **Edit Behaviours** icon to display the **Behaviour and Bindings Editor** for the **Tile**. See Figure 363.

Behaviours tab

To *deselect* all widgets, select the **Tile** by clicking in **Tile Editor** background

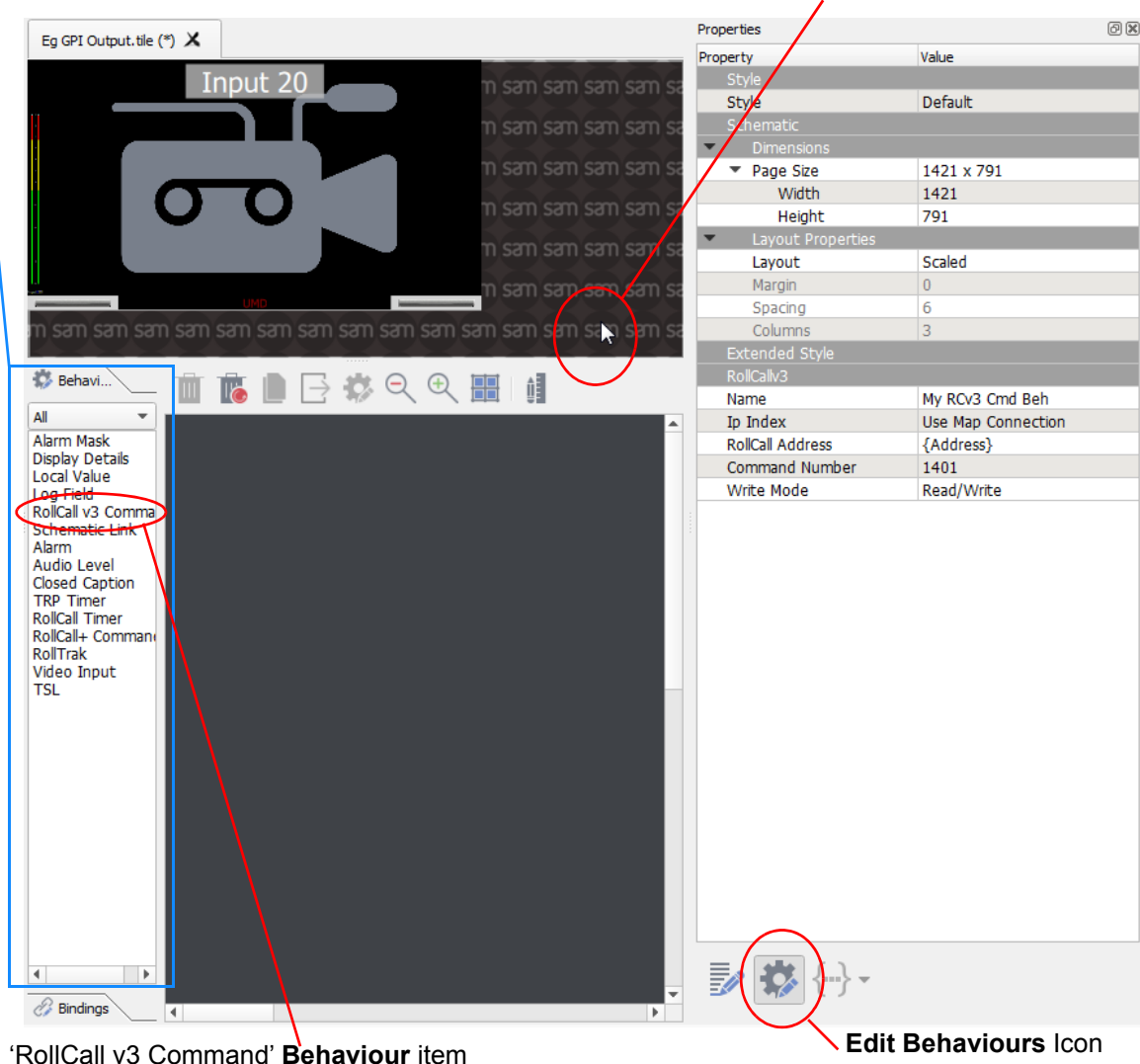


Figure 363 Tile Behaviour and Bindings Editor

4. Click on the 'RollCall v3 Command' name in the **Behaviours** tab. A **RollCall v3 Command Behaviour** block is added to the editor area.
5. Double-click on the **RollCall v3 Command Behaviour** block. The properties dialog for the selected **Behaviour** is shown.

6. Modify the **Behaviour**'s properties to the values given in Table 29.

| Property | Value | Comment |
|----------------|--------------------|---------------------------------|
| Name | My GPIO Cmd Beh | |
| IP Index | Use Map Connection | |
| Address | {Address} | |
| Command Number | 1401 | See Command Number note. |
| Write Mode | Read/Write | |

Table 29 Behaviour Property Values

Note:

Command Number:

"1401" is the RollCall command number for GPI 1.

Command numbers are:

1401 for GPI 1, 1402 for GPI 2, 1403 for GPI 3, and 1404 for GPI 4.

The completed properties dialog is shown in Figure 364.

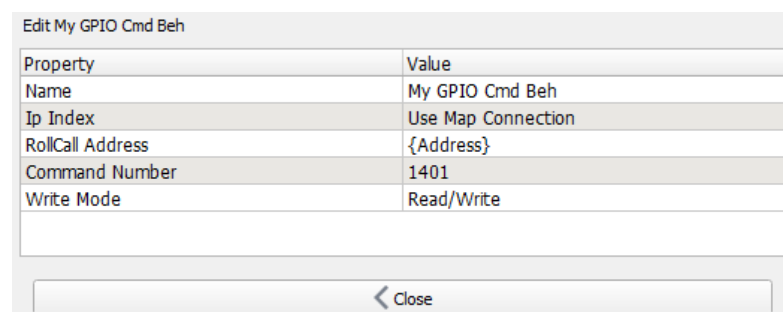


Figure 364 RollCall v3 Command Behaviour Properties Dialog

7. Click **Close**.

This has created a **Behaviour** to assert/de-assert the multiviewer's GPI 1 output.

8. In the **Tile Editor**, click outside the tile to deselect any widget.
A **Tile**'s properties are shown in the **Properties Box** when no widgets are selected.
The **Tile** properties are shown Figure 365; they now show some properties associated with the newly-added **RollCall v3 Command Behaviour**, 'My GPIO Cmd Beh'.

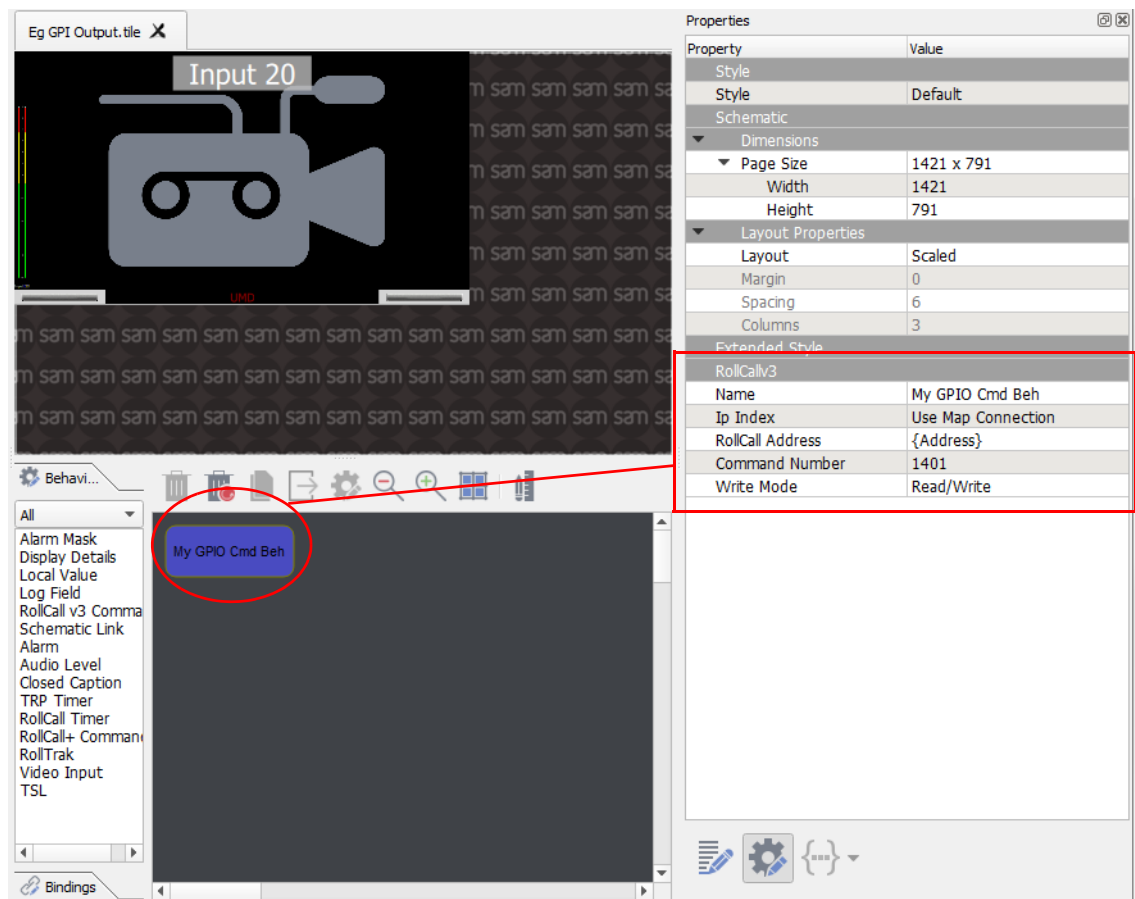


Figure 365 A Tile's New Properties, Associated with an Added Behaviour

9. Click Save.

This has created a new **Behaviour** on the **Tile** which can send a RollCall message to control GPI 1's output.

Now proceed to add a **Binding** to the **Tile**. This will get status information about our video input and process it to control the GPIO:

- 10.** Click on the 'Mapped Value' name in the **Bindings** tab.
A **Mapped Value Binding** block is added to the editor area.
- 11.** Double-click on the **Mapped Value Binding** block.
The properties dialog for the selected **Binding** is shown.
- 12.** Modify the **Binding's** properties to the values given in Table 30:

| Property | Value | Comment |
|------------------|-----------------|---|
| Name | My MappedV Bind | |
| Source Behaviour | Input State | This is a built-in Behaviour with a Video Widget. |
| Behaviour Value | State | See State Value note. |

Table 30 Binding Property Values

Note:**State Value in RollCall v3 Command Behaviour 'Input State':**

| State value | Meaning |
|-------------|----------------------|
| 0 | No alarm |
| 49 | Acknowledged Warning |
| 50 | Warning |
| 99 | Acknowledged Error |
| 100 | Error |

13. At **Bind Rules**, click **Add** twice.

Two blank **Bind Rule** entries appears in the **Bind Rules** table.

The **Bind Rules** must now be set up to process the **Behaviour** value, 'State':

14. Modify the new **Bind Rule** entry items; the completed **Bind Rules** are shown in Figure 366.

| Operator | Expression | Result |
|----------|------------|--------|
| Equals | 100 | 1 |
| Default | | 0 |

Buttons: Move Down, Move Up, Add, Delete

Result

Figure 366 Bind Rules

The **Result** of the **Bind Rules** now needs to be linked to a target **Behaviour** (generally, this may be a widget property value or some other **Behaviour**):

15. At **Target**, select **Behaviour** 'My GPIO Cmd Beh'.

Figure 367 shows the resulting completed **Bindings** properties dialog screen.

Edit My MappedV Bind

Common Properties

Name: My MappedV Bind

Source Behaviour: Input State [Select]

Behaviour Value: State [v]

Property to bind:

Bind Rules:

| Operator | Expression | Result |
|----------|------------|--------|
| Equals | 100 | 1 |
| Default | | 0 |

Buttons: Move Down, Move Up, Add, Delete

Target: My GPIO Cmd Beh [Select]

[Close]

Figure 367 Binding Properties

16. Click **Close**.

17. Click the **Update Layout** icon.

The **Behaviours** and **Bindings** for the **Tile** are shown. See Figure 368.

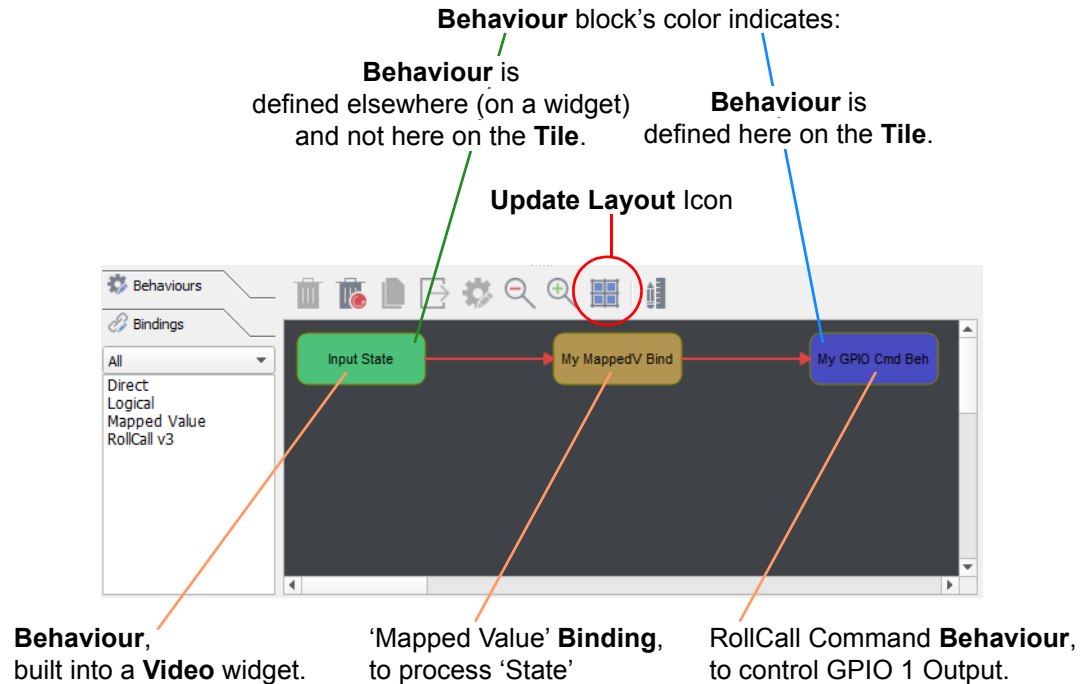


Figure 368 Tile Behaviours and Binding

18. Click **Save**.

19. Click **Save Project**.

This has created a **Binding** to process data from an existing **Behaviour** (called "Input State"). The **Binding** generates a data value from the video input state. Thus:

- "1" when the input state value indicates an error.
- "0" otherwise.

The data value is passed to a newly-created **Behaviour** ("GPIO1"), which asserts a GPI output on the multiviewer in the event of a video input error.

10.2.3 Auto-Tile and GPI Output

Use this section if you have an **Auto-Tile** that requires some GPI output functionality and do not wish to create a **Custom Tile**.

To configure an **Auto-Tile** to control a GPI output:

1. In the Orbit **Wall Editor**, select an **Auto-Tile** which contains a video widget.

The **Auto-Tile's** 'wall-level' properties are shown in the **Properties Box**, see Figure 369.

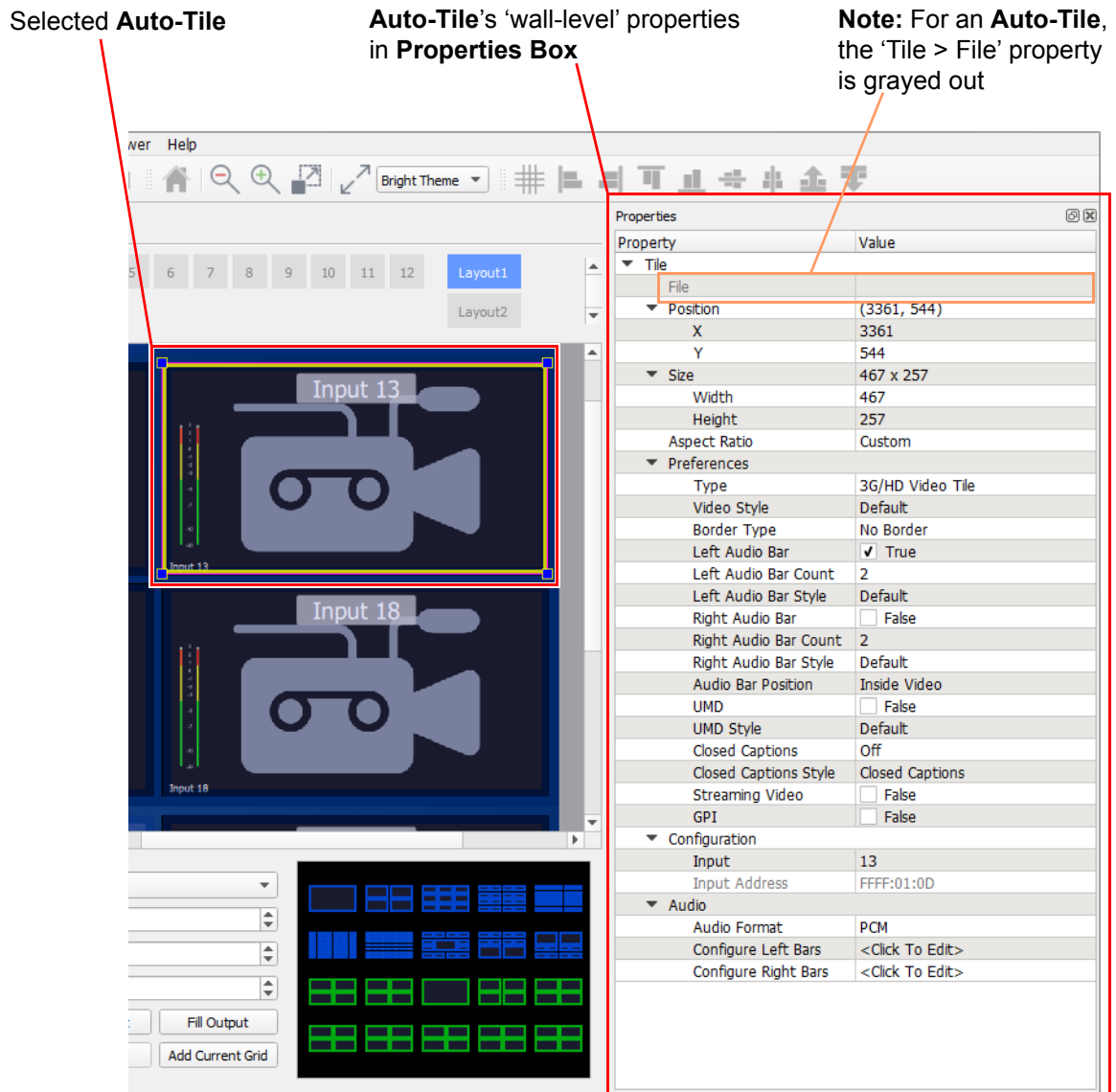
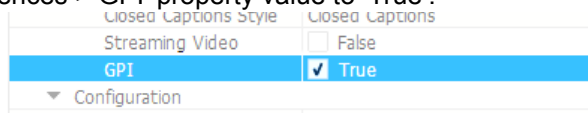


Figure 369 Auto-Tile 'Wall-Level' Properties

- Set the 'Preferences > GPI' property value to 'True'.



Extra GPI properties now appear. See Figure 370.

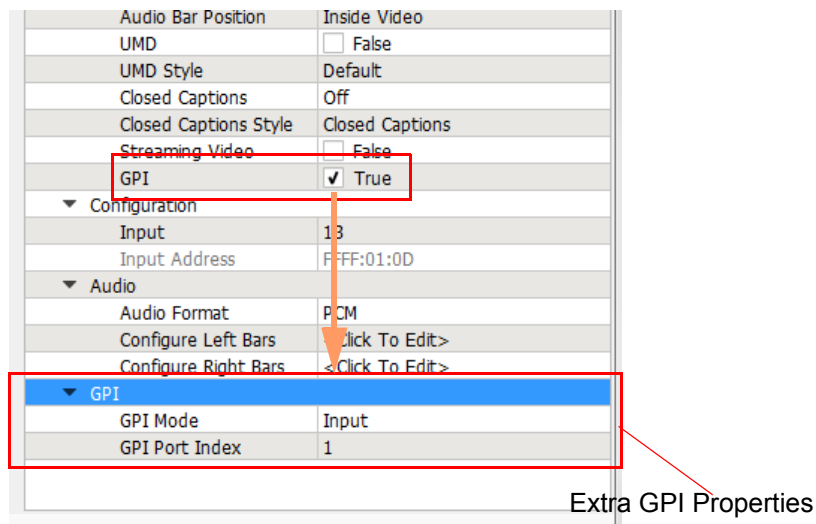


Figure 370 Extra GPI Properties for an Auto-Tile

- Modify the properties to the values given in Table 31:

| Property | Value |
|----------------|--------|
| GPI Mode | Output |
| GPI Port Index | 1 |

Table 31 New Property Values

- Click **Save**.
- Click **Save Project**.

This has configured the **Auto-Tile** to control GPI Output number 1.

Note: The **Auto-Tile** inherently already has the required **Behaviours** and **Bindings** set up to assert the GPI output when a video input error occurs.

10.3 On-Screen Video Input Status

This example explains how to add on-screen video input status information to a video input tile on a multiviewer video wall.

A video widget on a **Custom Tile** can be manually configured to display various extra pieces of information about its input video; a video tile is configured to show some lines of text in a text box. A **Behaviour** and a **Binding** can then be used to define exactly what information the lines of text will show, see Figure 371.

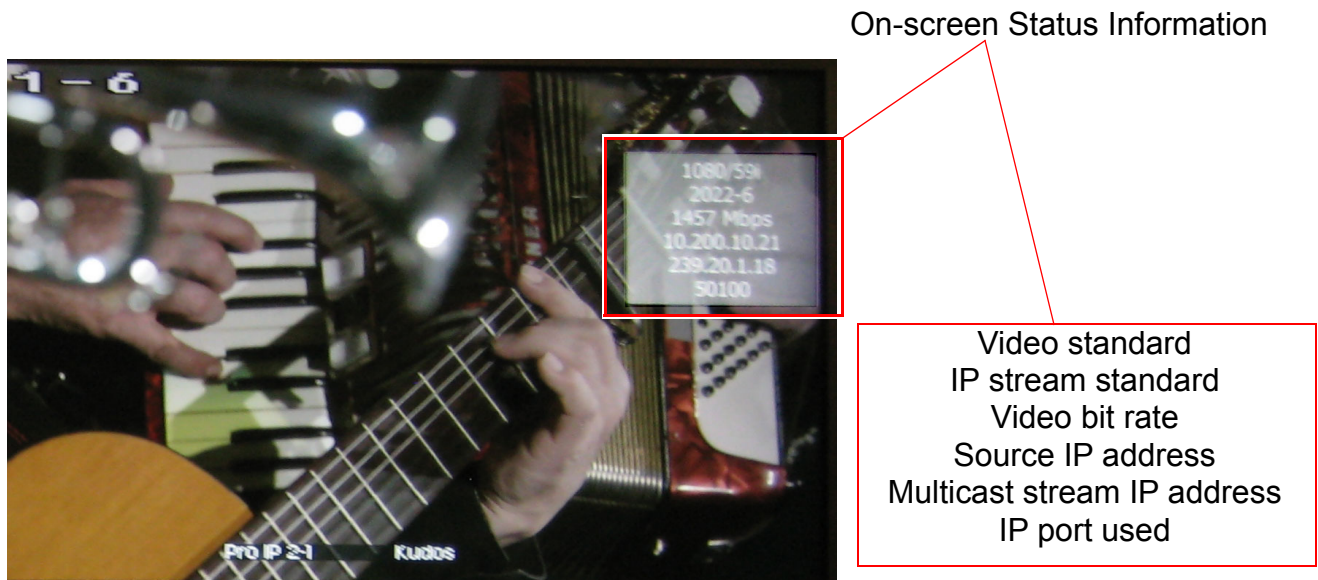


Figure 371 On-Screen Status Messages (IP Status Example Shown)

A video IP stream example is used, in Figure 371, which display information about a Video IP input stream for an IP Multiviewer. However, the same principles can be applied to SDI Video inputs on other Grass Valley MV-8 Series Multiviewers to display other status message information on-screen in a similar way.

For the case of an IP Multiviewer, video IP stream input information can be shown. For the displaying of other information or for other MV-8 series multiviewers, the procedure below can be adapted.

10.3.1 Create a Video Tile Style Showing Text Lines

Create a video tile style which shows some text lines:

1. Open the **Theme Editor** for a selected theme and view the video widget styles.

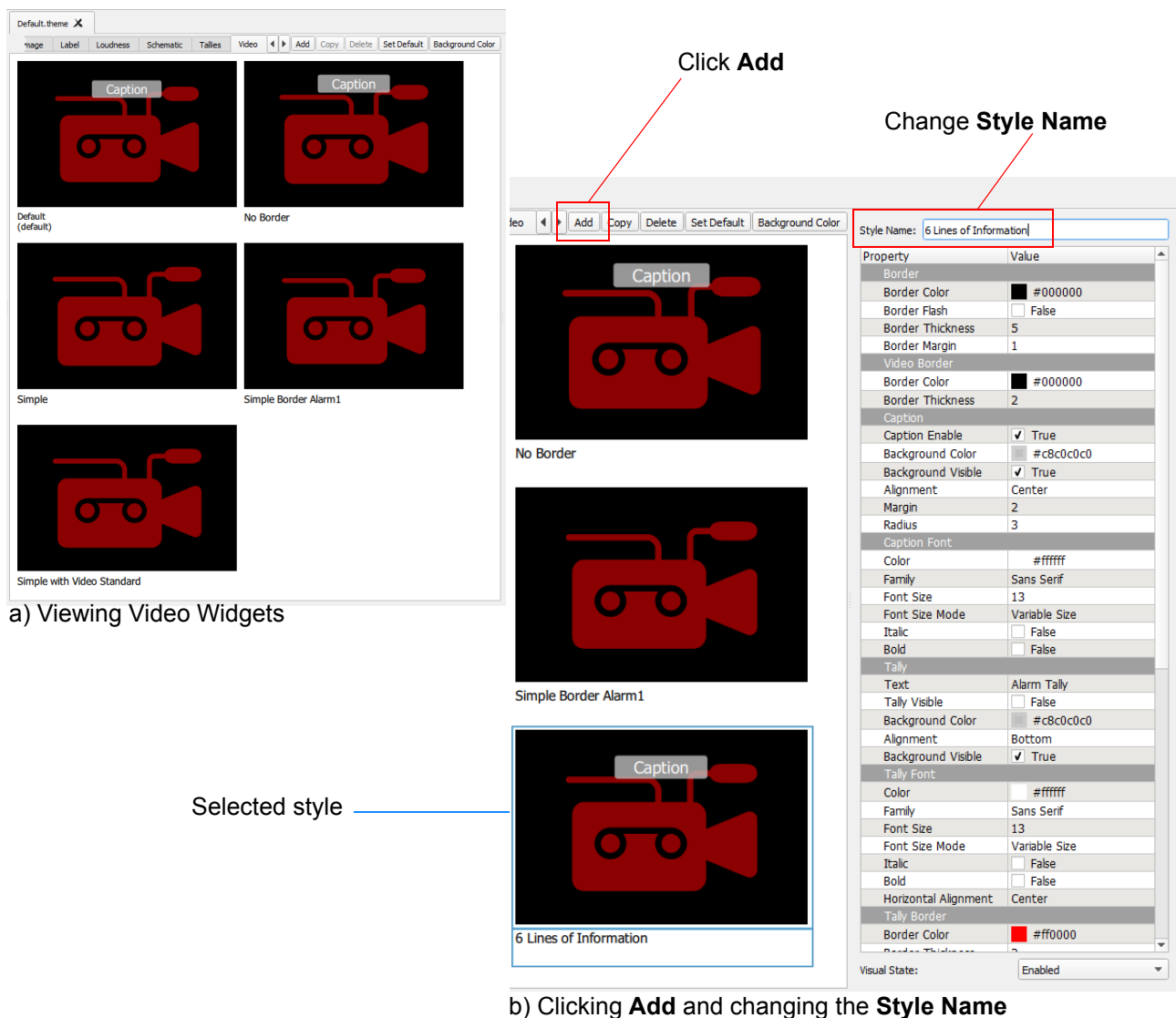


Figure 372 Video Widget Styles

2. Click **Add**.
A new video widget style appears.
3. Select the new widget style and change the **Style Name**. See Figure 372b.
 - **Style Name** '6 Lines of Information'
4. In the **Properties Box**, change the following properties:

| Property | New value |
|-----------------------|-----------|
| Status - Alignment | Right |
| Status - Status Lines | 6 |

The resulting video tile style now has a text box overlaying the video on the right-hand side. See Figure 373.

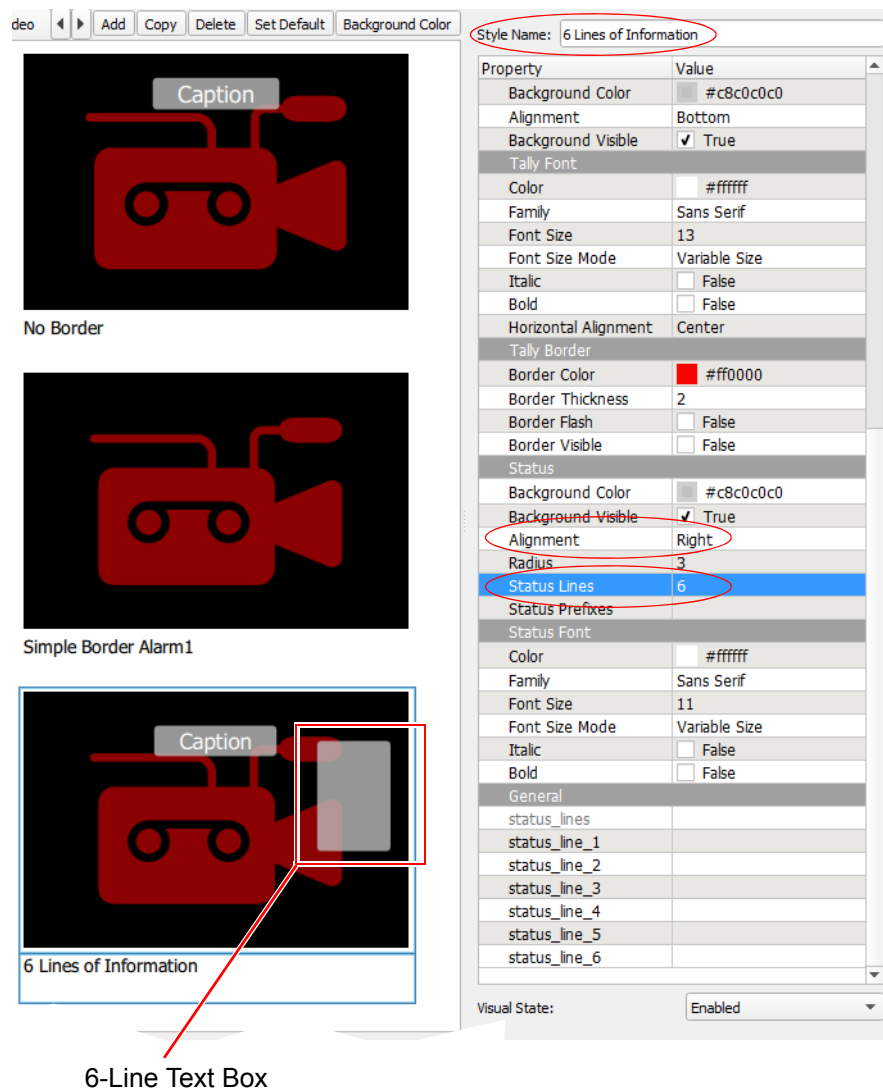


Figure 373 Video Widget Styles

5. Click the **Save File** icon to save this style change.

A video widget style called '6 Lines of Information' has been created.

10.3.2 Set Up the Number of On-Screen Text Status Lines

The status lines are shown on a video widget:

1. Edit a **Custom Tile** containing a video widget in the **Tile Editor**.
2. Select the **Video** widget.
The video widget's properties are shown.
3. Set the **Style** property to '6 Lines of Information'. See Figure 374.

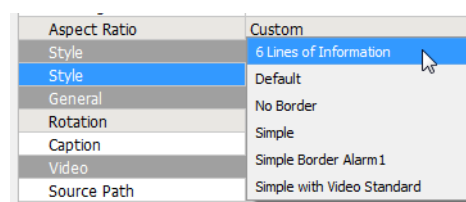


Figure 374 Set Style Property

Note:

If the **Video** widget has been part of an **Auto-Tile**, then:

- Some **Behaviours** will have already been set up.
- In particular, a **Log Field Behaviour** for some log field information, which results in some 'Log Field Behaviour' properties being shown in the **Tile Editor's Properties Box**. See Figure 375.

If the **Video** widget has been newly-added to the tile, then it is likely that no such **Behaviour** is set up.

Video widget properties

The screenshot shows the 'Eg OnScreen Status.tile' window with a video widget selected. The widget is labeled 'Input 24' and has a camera icon. The Properties panel on the right shows the following settings:

| Property | Value |
|------------------------------|--|
| Dimensions | |
| Position | (0, 0) |
| X | 0 |
| Y | 0 |
| Size | |
| Width | 467.00 |
| Height | 257.00 |
| Aspect Ratio | Custom |
| Style | |
| Style | Default |
| General | |
| Caption | Input 24 |
| Video | |
| Source Path | |
| Video Aspect Ratio | Auto |
| Muted | <input checked="" type="checkbox"/> True |
| Paused | <input type="checkbox"/> False |
| Deinterlace | On |
| Extended Style | |
| Video Input Behaviour | |
| Name | Video Input |
| RollCall Address | {Address} |
| Input | {Input} |
| Log Field Behaviour | |
| Name | Input Standard Log |
| RollCall Address | {Input Address} |
| Number of Headers | 1 |
| Headers | |
| Alarm Behaviour | |
| Name | Input State |
| Mode | Single Address |
| RollCall Address | {Input Address} |
| Configuration | <Click To Edit> |

Figure 375 Video Widget Selected and Properties

4. Select the **Video** widget and click the **Edit Behaviours** icon. The **Behaviours and Bindings Editor** is shown for the selected widget. See Figure 376.

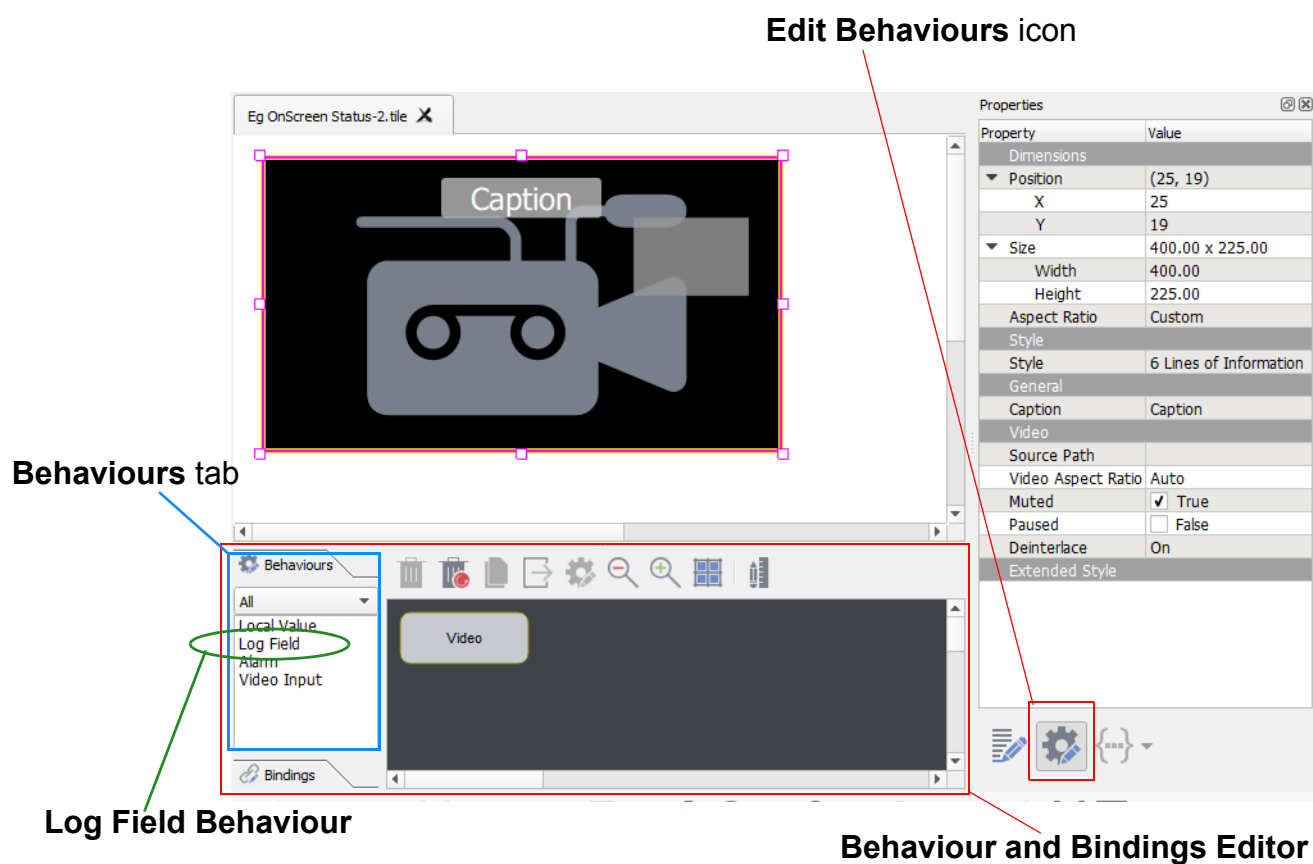
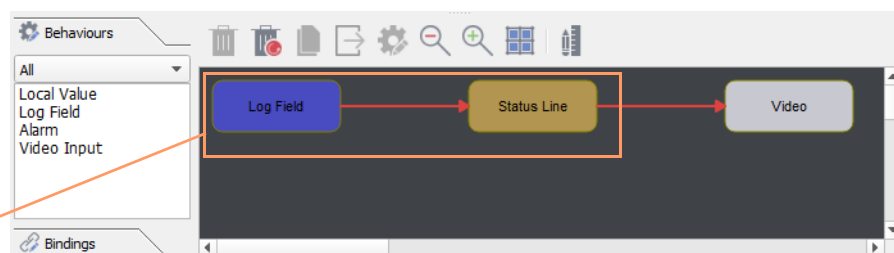


Figure 376 Behaviour and Bindings Editor

- Click on the 'Log Field' name in the **Behaviours** tab.
A **Log Field Behaviour** block and a **Binding** block are added to the editor screen.
See Figure 377 a) and b).

a) For a newly-added **Video** widget

Log Field Behaviour block
+ **Status Line Binding** block

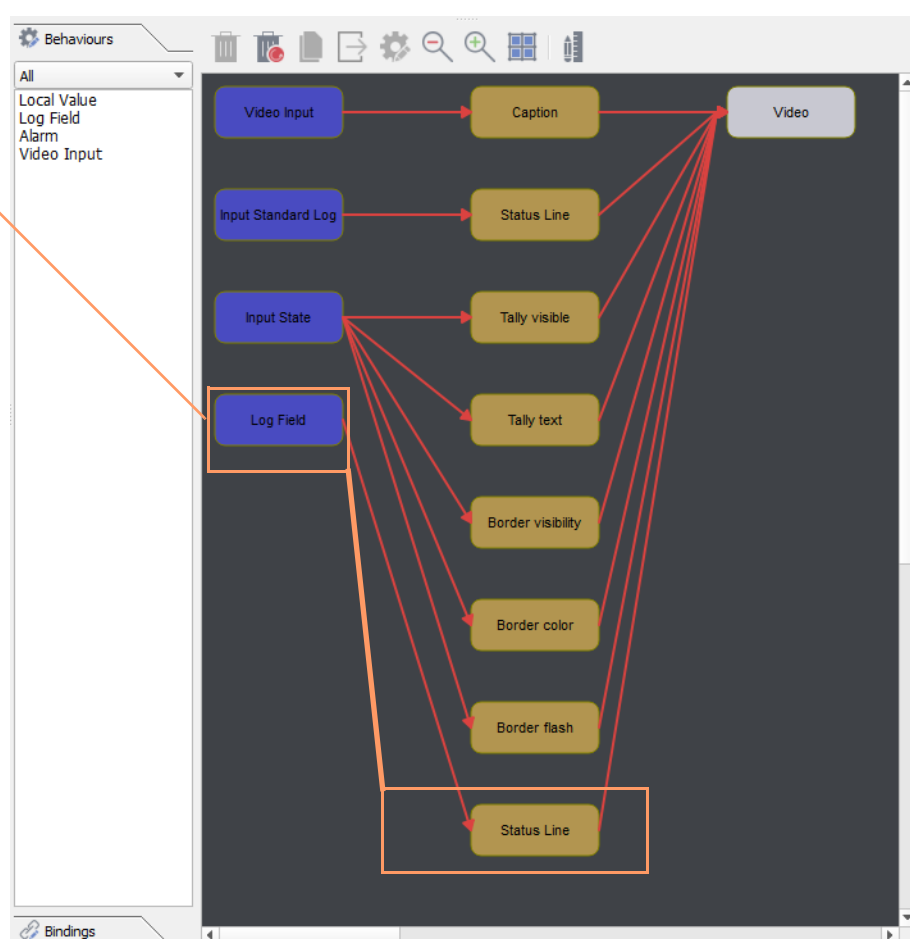
b) For a **Video** widget from an **Auto-Tile**

Figure 377 Behaviour and Binding Blocks

6. Double-click on the **Log Field Behaviour** block.
The **Behaviour** properties dialog is shown.

7. Change the following widget 'Log Field Behaviour' properties:

| Property | New Value |
|--------------------------|------------------|
| Name | My Log Field Beh |
| RollCall Address | {Address} |
| Number of Headers | 1 |

Table 32 Log Field Behaviour Property Value Changes

8. Click **Close**.
9. Click **Save**.

A **Log Field Behaviour** has been added to the **Video** widget.

| | |
|---------------------|--------------------------------|
| Paused | <input type="checkbox"/> False |
| Deinterlace | On |
| Extended Style | |
| Log Field Behaviour | |
| Name | My Log Field Beh |
| RollCall Address | {Address} |
| Number of Headers | 1 |
| Headers | |

Our **Video** widget now has some **Log Field Behaviour** properties whether the widget was from an **Auto-Tile** or the widget is freshly-added to the tile.

The next change modifies a **Behaviour** property and may be done either:

- through the **Behaviours and Bindings Editor**,
- or directly in the widget's **Properties Box**.

The **Properties Box** method is described:

10. View the **Video** widget's properties in the **Tile Editor's Properties Box**.
11. Change the following widget 'Log Field Behaviour' property:

| Property | New Value |
|--------------------------|-----------|
| Number of Headers | 6 |

12. Click the **Save** icon in the main tool bar.

To change the **Headers** property:

13. Click inside the (blank) **Headers** 'Value' cell to display a "... " button. See Figure 378.

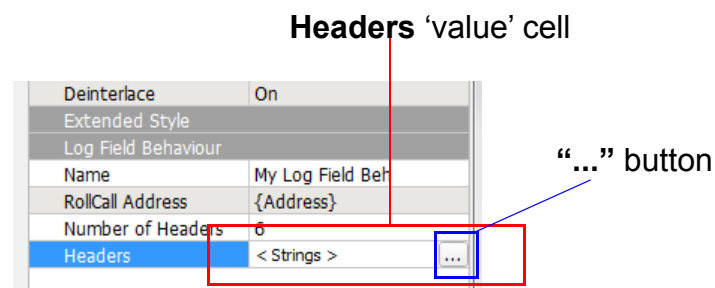
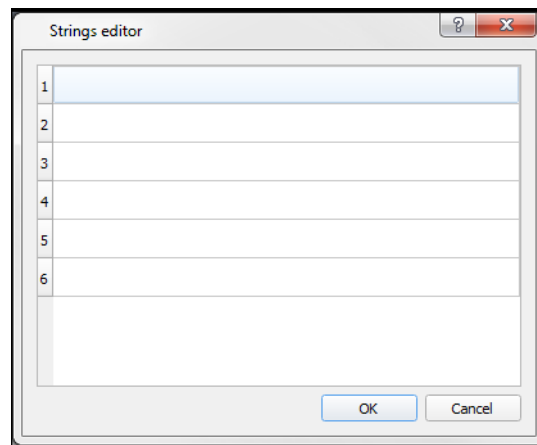


Figure 378 Headers Value

14. Click the “...” button, see Figure 378.
The **Strings Editor** dialog is shown with six blank lines. See Figure 379a.

a) **Strings Editor** dialog

| | |
|---|--------------------------|
| 1 | INPUT_{Input}_STANDARD |
| 2 | INPUT_{Input}_VIDENC |
| 3 | INPUT_{Input}_BITRATE |
| 4 | INPUT_{Input}_SOURCEADDR |
| 5 | INPUT_{Input}_MCASTADDR |
| 6 | INPUT_{Input}_DSTPORT |

b) Strings Editor table entries

See **Strings Editor** Note.

Figure 379 Strings Editor: a) Editor dialog. b) Table Entries

15. Enter the **Strings Editor** values shown in Figure 379b.

Note:

Strings Editor Entries:

The **Strings Editor** entries are log field names.
The entry “INPUT_{Input}_STANDARD”, in Figure 379b, applies to SDI or IP stream video inputs. The other table entries are for IP inputs only.

Generally, the table entries in Figure 379b may be adapted to display other status information from log fields or from other MV-8 series multiviewers.
For information about a multiviewer’s strings, right-click on the device in the Orbit Network View pane and select ‘details’ to view the live log fields.

The completed **Strings Editor** dialog is shown in Figure 380.

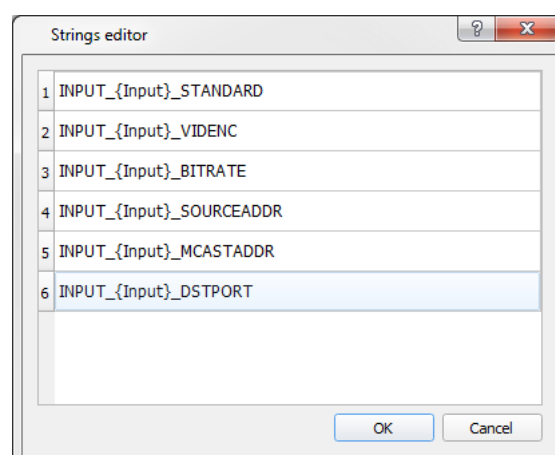


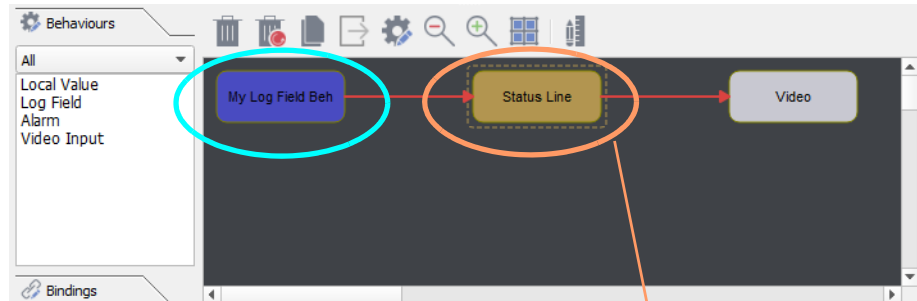
Figure 380 Strings Editor

16. Click **OK** to close the **Strings Editor**.
17. Click the **Save** icon in the main tool bar.

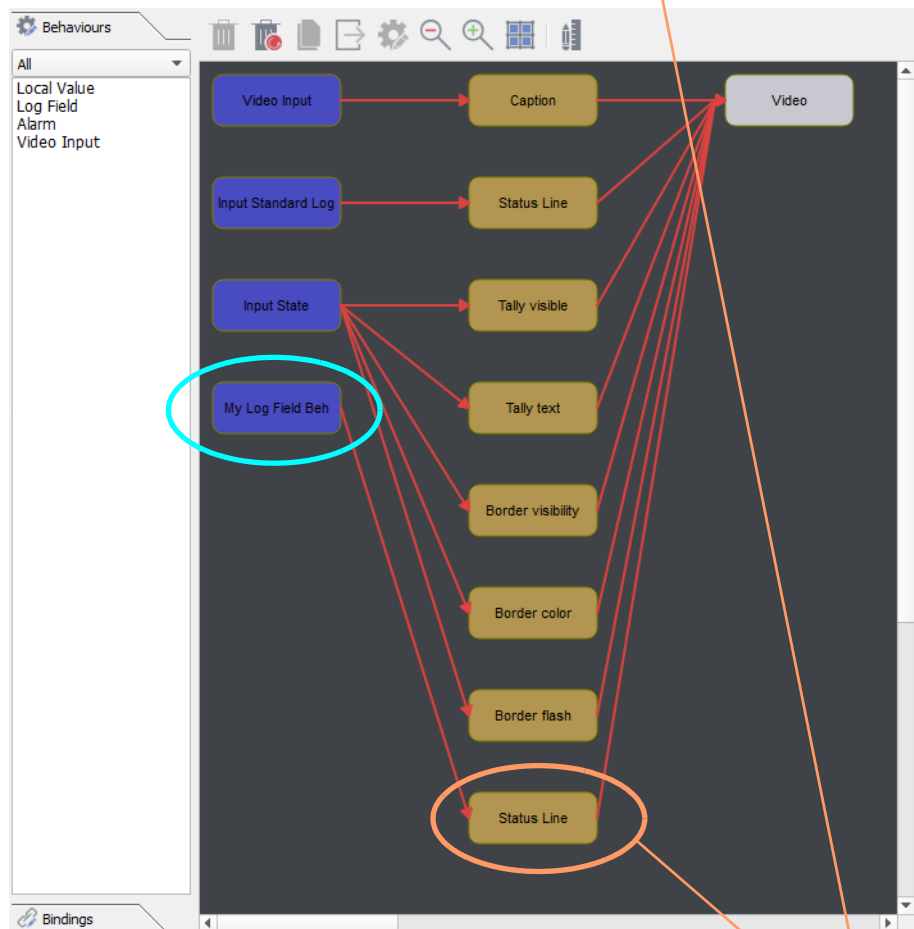
The **Log Field Behaviour** has been modified to get certain log field data values about a multiviewer input.

Next, link the log field data values to on-screen fields:

18. With the video widget still selected, click the **Edit Behaviours** icon. The **Behaviours and Bindings Editor** is shown for the selected widget. See Figure 381.



a) **Property Binding** for video widget status lines.



b) **Behaviours and Bindings** for a **Video** widget from an **Auto-Tile**. **Property Binding** for video widget status lines highlighted.

Property Binding for video widget status lines.

Figure 381 Property Bindings for Video Widget Status Lines

19. Double-click the **Property Binding** block (highlighted in Figure 381). The **Bindings** property dialog is shown for the selected **Binding**.

Next, the **Binding** block is configured to send status text to the first status line of the Video widget by editing the **Binding** properties:

20. Change the **Binding** properties to values shown in Table 33.

| Property | New Value | Comment |
|-------------------------|------------------------|-------------------------|
| Name | My Line 1 Bind | |
| Source Behaviour | My Log Field Beh | |
| Behaviour Value | INPUT_{Input}_STANDARD | |
| Property to Bind | status_line_1 | General > status_line_1 |

Table 33 Property Binding Property Value Changes

The completed dialog is shown in Figure 382.

Figure 382 New Property Binding and its “Binding Properties”

21. Click **Close**.
22. Click **Save**.

This has created a **Binding** to link the video input standard value from **Behaviour** ‘My Log Field Beh’ to line 1 of the video widget’s status box.

Now, the remaining lines of the (6-line) on-screen status box need to be connected up in a similar way:

23. Click on the **Property** name in the **Bindings** list to create a further **Property Binding** block on the **Behaviour and Bindings Editor** editor screen.
24. Double-click on the new **Property Binding** block to view the properties dialog.
25. Change the **Binding** properties to values shown in Table 34.

| Property | New Value | Comment |
|-------------------------|----------------------|-------------------------|
| Name | My Line 2 Bind | |
| Source Behaviour | My Log Field Beh | |
| Behaviour Value | INPUT_{Input}_VIDENC | |
| Property to Bind | status_line_2 | General > status_line_2 |

Table 34 Property Value Changes for second Property Binding Block

26. Click **Close**.

27. Click **Save**.

This has created a **Binding** to link the video encode type value from **Behaviour** 'My Log Field Beh' to line 2 of the video widget's on-screen status box.

28. Click on the **Property** name in the **Bindings** list

to create a further four **Property Binding** blocks on the **Behaviour and Bindings Editor** editor screen.

29. Double-click on each new **Property Binding** block to view the properties dialog.

30. Change the **Binding** properties of each of the Binding blocks to the values shown in Table 35.

| Property | New Value for 3rd Binding Block | New Value for 4th Binding Block | New Value for 5th Binding Block | New Value for 6th Binding Block |
|-------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|
| Name | My Line 3 Bind | My Line 4 Bind | My Line 5 Bind | My Line 6 Bind |
| Source Behaviour | My Log Field Beh | My Log Field Beh | My Log Field Beh | My Log Field Beh |
| Behaviour Value | INPUT_{Input}_BITRATE | INPUT_{Input}_SOURCEADDR | INPUT_{Input}_MCASTADDR | INPUT_{Input}_DSTPORT |
| Property to Bind | status_line_3 | status_line_4 | status_line_5 | status_line_6 |

Table 35 Property Value Changes for second Property Binding Block

31. Click **Close**.

32. Click **Save**.

Overall this has created six **Binding** blocks to link the status data from **Behaviour** 'My Log Field Beh' to six lines of the video widget's on-screen status box. See Figure 383.

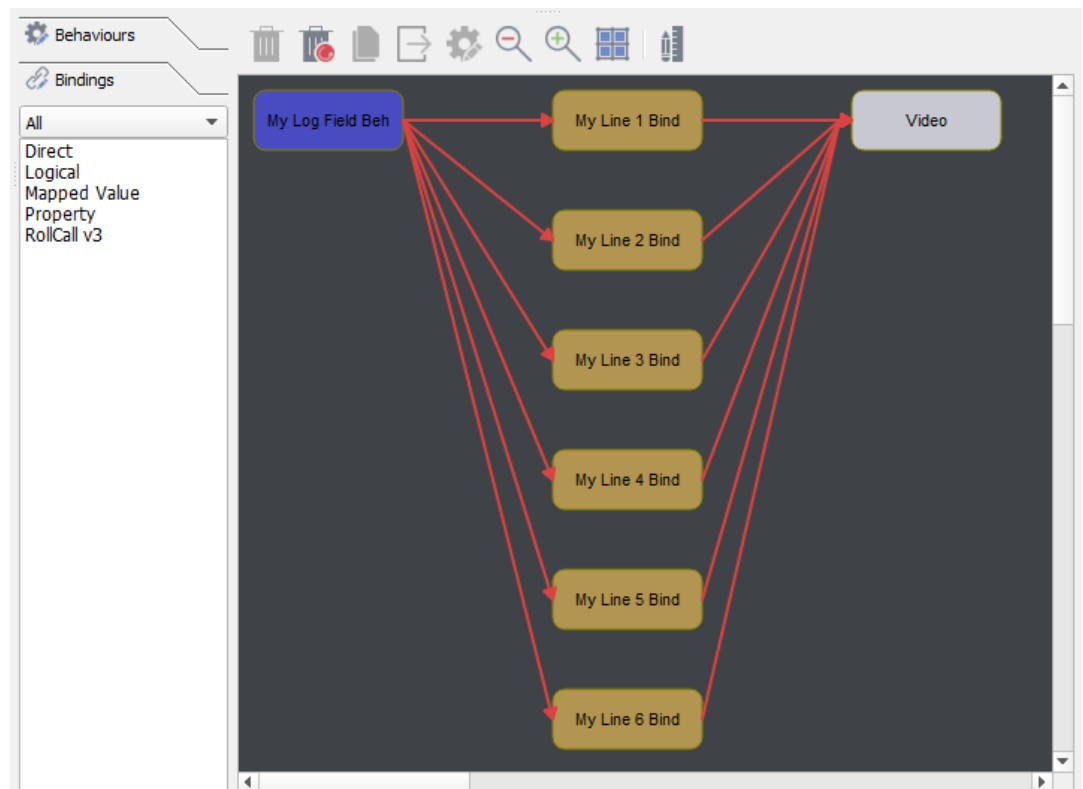


Figure 383 Six New Property Bindings

For the case of a video widget that has come from an **Auto-Tile**, there are more pre-existing **Behaviours** and **Bindings**, see Figure 384.

Six **Bindings** handling data to the Video widget status lines.

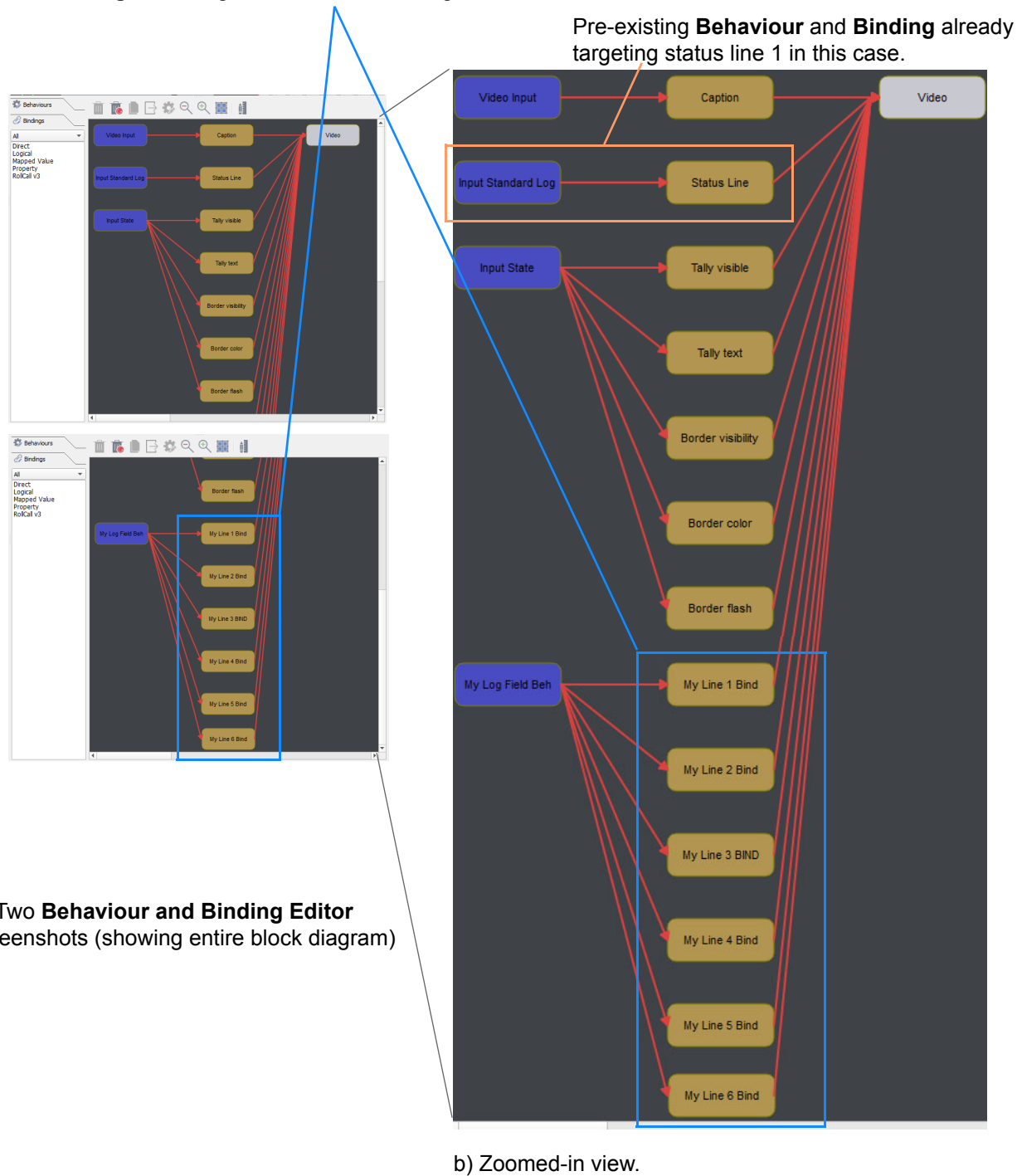


Figure 384 Six New Property Bindings

For the case in Figure 384, there are two pre-existing **Behaviour** and **Binding** items that may already be present controlling line 1. These are highlighted in Figure 384 and require deleting for our example.

33. Select and delete the pre-existing **Behaviour** and **Binding** highlighted in Figure 384.
34. Click **Save**.

A Terminology

Terminology

| | |
|------------------------------------|-----|
| Terminology for Multiviewers | 319 |
|------------------------------------|-----|

A.1 Terminology for Multiviewers

This section describes the terminology used when discussing multiviewers and video walls. It should be read alongside the example dual-screen video wall illustration of Figure 385.

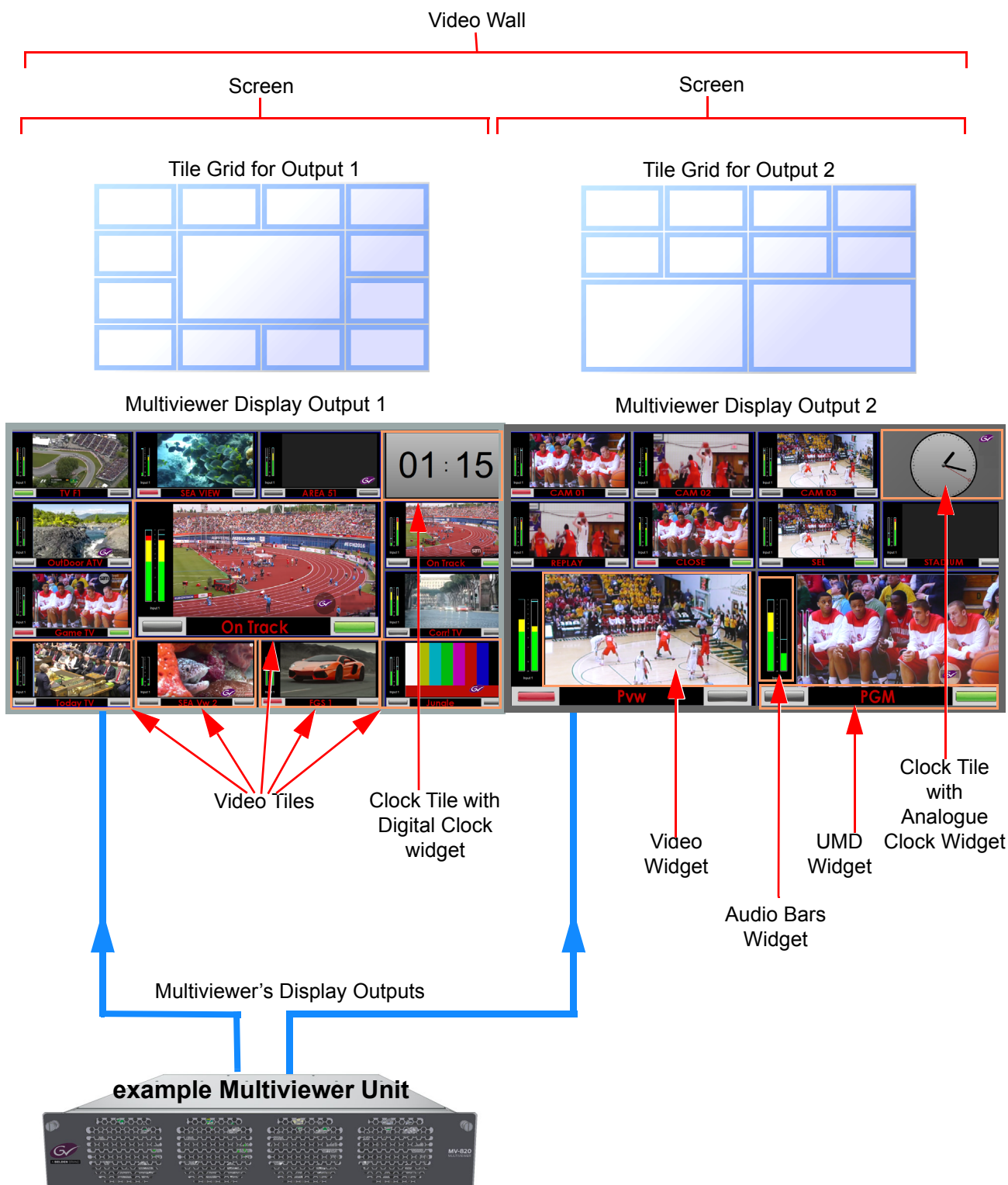


Figure 385 Example Video Wall - Made up of Two Monitor Display Screens

| Term | Definition |
|----------------------------------|---|
| Wall, Video Wall | One or more monitor display screens configured to form one large screen area. |
| Screen | Display area of one individual monitor/display device. |
| Tile Grid | A wall/screen is divided up into rectangular grid areas; various sizes and arrangements are possible. Information is displayed in each rectangular area. Used for quick-positioning of tiles onto a wall. |
| Fine Grid | A fine grid is used for the fine-positioning of graphical elements in Orbit; for example, for tiles on a wall or widgets on a tile. |
| Tile | A rectangular area on a screen, usually displaying video inputs and other supporting information. For example, a tile may display a video picture with audio bars and other related status information. Other information may be displayed in tiles, for example, time, images, labels or text. |
| Widget | Widgets are graphical on-screen elements which populate tiles. There can be one or more widgets per tile. Widgets display information, for example, audio sound level, time, video picture, text labels, tally information. Widgets include: Audio bars, Clocks, Images, Labels, Tally Lamps, Timers, UMD's and Video. |
| Widget Style | Widgets can have differently-styled versions, providing a different look and feel. A widget style modifies the on-screen appearance of a widget but does not change a widget's function. For example, the Audio Bars widget may be defined with left-handed and right-handed styles. |
| Theme | A Theme is a set of widget styles which can be applied to a project. Each widget may have more than one style. |
| Multiviewer Input | A Multiviewer has many video inputs. For example, a MV-820 Multiviewer has up to 48 multiviewer inputs. These multiviewer inputs are processed by the Multiviewer and appear in video wall tiles. |
| Display Output | An output signal from a multiviewer carrying all or part of a video wall image. The output can be SDI (Coax or Fiber) or HDMI or a Video IP stream, depending on the multiviewer model. For example, a 1080P SDI output from a MV-820 Multiviewer. Each display output is connected to a monitor display that forms all or part of a multiviewer video wall. |
| Streamed Out Video Inputs | On some multiviewer models, multiviewer inputs are streamed out from a Grass Valley Multiviewer as scaled-down, H.264-compressed IP streams. These are outputs from the multiviewer. These are available for viewing on a PC desktop (via H.264 stream viewing software, for example, Orbit with a MV-800-DT license option). Note: These are compressed H.264 IP streams. These are <i>not</i> uncompressed video IP streams nor VC-2 video IP streams. |

Table 36 Definition of Terms

B List of Widgets

List of Widgets

| | |
|-----------------------------|-----|
| Widget Tool Bar Items | 322 |
|-----------------------------|-----|

The **Tile Editor** allows various graphical widget objects to be dragged onto the tile being designed. All the available widgets and graphical objects are on the **Widgets Tool Bar** in the **Tile Editor**, see Figure 386.



Figure 386 Widgets Tool Bar

B.1 Widget Tool Bar Items

Table 37 briefly describes the graphical widget objects. Widgets are listed in the same order as they appear on the **Widgets Tool Bar**:



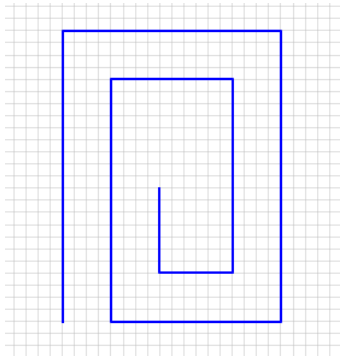



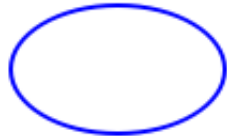

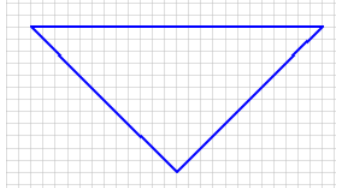

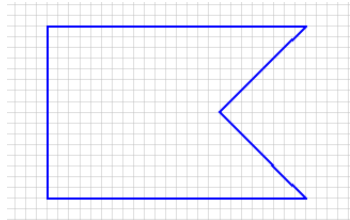
| Graphical Object | Description | On-Screen Example |
|--|---|---|
|  <p>'Shape' widgets:</p> | | |
|  Line | <p>Click the Line icon to enter 'line drawing' mode. Straight line segments can be drawn on the tile or schematic:</p> <p>Click to begin to draw a line; Move cursor and click to form corners.</p> <p>Press ESC to complete the line drawing, deselect the tool and exit 'line drawing' mode.</p> |  |
|  Rectangle | <p>Click the Rectangle icon for 'rectangle drawing' mode.</p> <p>A rectangle can be drawn:</p> <p>Click and hold to begin to draw. Drag to draw the rectangle. Release to finish drawing.</p> |  |
|  Ellipse | <p>Click the Ellipse icon for 'ellipse drawing' mode.</p> <p>An ellipse can be drawn:</p> <p>Click and hold to begin to draw. Drag to draw the ellipse. Release to finish drawing.</p> |  |
|  Triangle | <p>Click the Triangle icon for 'triangle drawing' mode.</p> <p>A triangle can be drawn:</p> <p>Click-and-drag to begin to draw. Release-click at second vertex. Release-click at third vertex to finish.</p> |  |
|  Polygon | <p>Click the Polygon icon for 'polygon drawing' mode.</p> <p>Straight line segments can be drawn.</p> <p>Click to begin to draw a line; click to add a corner and complete each line segment.</p> <p>Right-click to finish drawing and deselect the tool.</p> |  |

Table 37 Widgets Tool Bar Items



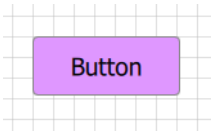

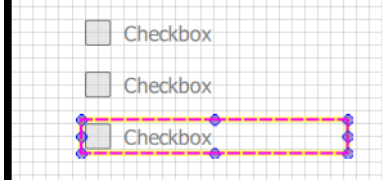
| Graphical Object | Description | On-Screen Example |
|---|--|---|
|  | | |
| <p>'Non-Shape' widgets:</p> | <p>To add these non-shape widgets, carry out the following mouse/key board action:</p> <p>To add a single widget:</p> <ul style="list-style-type: none"> •Click once on the tile. <p>To add multiple widgets:</p> <ul style="list-style-type: none"> •Hold down the 'Ctrl' key and click to add each widget. •Release the 'Ctrl' key and click to place the last widget. | |
|  <p>Button</p> | <p>Click the Button icon for 'button drawing' mode.</p> <p>Move cursor onto tile/schematic with a Button widget on the cursor.</p> <p>To add a single widget:</p> <ul style="list-style-type: none"> • Click once on the tile. <p>To add multiple widgets:</p> <ul style="list-style-type: none"> • Hold down the 'Ctrl' key and click to add each widget. • Release the 'Ctrl' key and click to place the last widget. |  |
|  <p>Check box</p> | <p>Click the Check Box icon to add a check box widget.</p> <p>The widget is attached to the cursor.</p> <p>To add a single widget:</p> <ul style="list-style-type: none"> • Click once on the tile. <p>To add multiple widgets:</p> <ul style="list-style-type: none"> • Hold down the 'Ctrl' key and click to add each widget. • Release the 'Ctrl' key and click to place the last widget. |  |

Table 37 Widgets Tool Bar Items (Continued)




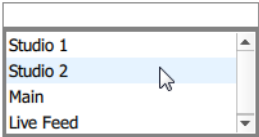

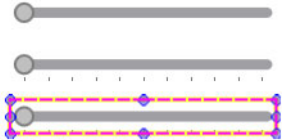
| Graphical Object | Description | On-Screen Example |
|--|---|---|
|  | | |
| <div><div>Combo box</div></div> | <p>Click the Combo Box icon to add a combo box widget.</p> <p>The widget is attached to the cursor. Click to drop the widget onto the tile.</p> |  |
| | <p>List of drop-down items: The list of drop-down items for the combo box is set in the widget properties, entered into the Properties Box.</p> <p>Set the General > Data property to a semi-colon separated list of data entries.</p> <p>For example, “Studio 1;Studio 2;Main;Live Feed”.</p> |  |
| <div><div>Slider</div></div> | <p>Click the Slider icon to add a slider widget.</p> <p>A single slider widget is attached to the cursor.</p> <p>To add a single widget:</p> <ul style="list-style-type: none">Click once on the tile. <p>To add multiple widgets:</p> <ul style="list-style-type: none">Hold down the ‘Ctrl’ key and click to add each widget.Release the ‘Ctrl’ key and click to place the last widget. |  |

Table 37 Widgets Tool Bar Items (Continued)

| Graphical Object | Description | On-Screen Example |
|------------------|--|-------------------|
| | | |
| | <p>Click the Text Edit icon to add a text edit widget.</p> <p>The widget is attached to the cursor. Click to drop the widget onto the tile.</p> | |
| | <p>Click the Group Box icon to add a Group Box widget.</p> <p>The group box widget is attached to the cursor. Click to drop the widget onto the tile.</p> | |
| | <p>Click the Label icon to add a text label widget.</p> <p>The widget is attached to the cursor.</p> <p>To add a single widget:</p> <ul style="list-style-type: none">Click once on the tile. <p>To add multiple widgets:</p> <ul style="list-style-type: none">Hold down the 'Ctrl' key and click to add each widget.Release the 'Ctrl' key and click to place the last widget. | |
| | <p>Click the icon to add a List Box widget.</p> <p>The widget is attached to the cursor. Click to drop the widget onto the tile.</p> | |

Table 37 Widgets Tool Bar Items (Continued)


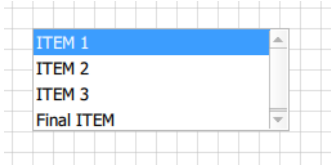

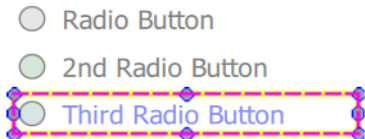


| Graphical Object | Description | On-Screen Example |
|---|--|---|
|  | <p>List of items: The list of items for the List Box should be entered in the Properties Box. Set the General > Data property to a semi-colon separated list of data entries. For example, "ITEM 1;ITEM 2;ITEM 3;...".</p> |  |
|  Radio button | <p>Click the icon to add a Radio Button widget.</p> <p>A single radio button widget is attached to the cursor. Click to drop the widget onto the tile.</p> <p>To add a single widget:</p> <ul style="list-style-type: none">Click once on the tile. <p>To add multiple widgets:</p> <ul style="list-style-type: none">Hold down the 'Ctrl' key and click to add each widget.Release the 'Ctrl' key and click to place the last widget. |  |
|  Image | <p>Click the Image icon to add an image widget.</p> <p>A single widget is attached to the cursor. Click to drop the widget onto the tile.</p> <p>Resize to suit.</p> <p>Set the file name and path of the (still) image to be displayed, extended property Image > Source Path.</p> <p>Set how the image fills the widget, extended property Image > Fill Mode</p> |  |

Table 37 Widgets Tool Bar Items (Continued)



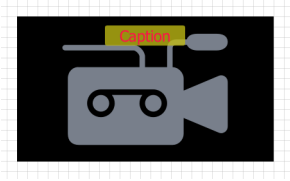

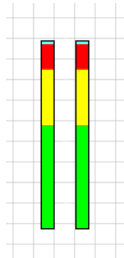


| Graphical Object | Description | On-Screen Example |
|---|--|---|
|  | | |
|  Video Stream | <p>Click the Video Stream icon to add a Video widget.</p> <p>A single widget is attached to the cursor. Click to drop the widget onto the tile.</p> <p>This widget is used in video Auto-Tiles and may show multiviewer video inputs or video from a received video IP stream.</p> <p>Set the video stream path with property Video > Source Path.</p> |  |
|  Audio Bars | <p>Click on the Audio Bars tool icon to add an audio bars widget.</p> <p>An audio bars widget is attached to the cursor.</p> <p>Click to drop the widget onto the tile.</p> <p>By default it is a pair of channels, but this is configurable in the properties, Audio > Number of Bars.</p> |  |
|  Clock | <p>Click the Clock icon to add a clock widget.</p> <p>The widget is attached to the cursor. Click to drop the widget onto the tile.</p> |  |

Table 37 Widgets Tool Bar Items (Continued)





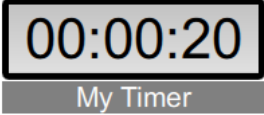
| Graphical Object | Description | On-Screen Example |
|--|---|---|
|  | | |
|  <div>Digital Clock</div> | <p>Click the Digital Clock icon to add a digital clock widget.</p> <p>The widget is attached to the cursor. Click to drop the widget onto the tile.</p> |  |
|  <div>Timer</div> | <p>Click the Timer icon to add a timer widget.</p> <p>The widget is attached to the cursor.</p> <p>To add a single widget:</p> <ul style="list-style-type: none">Click once on the tile. <p>To add multiple widgets:</p> <ul style="list-style-type: none">Hold down the 'Ctrl' key and click to add each widget.Release the 'Ctrl' key and click to place the last widget. |  |

Table 37 Widgets Tool Bar Items (Continued)

| Graphical Object | Description | On-Screen Example |
|--|--|--|
|  | | |
| <div>Component</div> | <p>Click the Component icon to embed another Orbit component object into another Orbit tile or schematic.</p> <p>Select a component from the Select File dialog which appears.</p> <div data-bbox="531 544 971 954"></div> <p>Components use widgets, like Tiles, and they are typically used when using Orbit to make Grass Valley Orbit MapView screens.</p> <p>Note: This is typically not a Multiviewer widget.</p> <p>Schematics (similar to tiles but a .schx file) can be designed to show status and hold controls. These can be added onto other schematics, or onto tiles, using the Component widget. Component editing is done in a similar way to Tile editing.</p> | |
| <div>Web</div> | <p>Click the Web icon to embed a web page into a tile (or into an Orbit schematic).</p> <p>Set the web page to be displayed path the URL property value.</p> | <div data-bbox="1077 1368 1469 1603"></div> |

Table 37 Widgets Tool Bar Items (Continued)





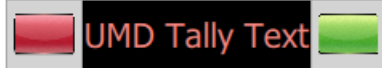



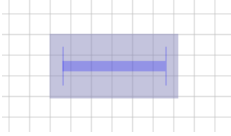
| Graphical Object | Description | On-Screen Example |
|--|--|---|
|  | | |
|  Tally Lamp | <p>Click the Tally Lamp icon to add a tally lamp widget.</p> <p>The widget is attached to the cursor. Click to drop the widget onto the tile.</p> <p>Default lamp color is set via the Left Lamp Color, Right Lamp Color and Text property value.</p> <p>Typically, the lamp color is required to respond to changes in system state. This is done with Behaviours and Bindings.</p> |  |
|  Under Monitor Display | <p>Click the UMD icon to add a UMD widget.</p> <p>The widget is attached to the cursor. Click to drop the widget onto the tile.</p> <p>Default lamp colors and text is set via the Lamp Color property value.</p> <p>Typically, the UMD is required to respond to changes in system data, for example TSL data. This is done with Behaviours and Bindings.</p> |  |
|  Border | <p>Click the Border icon to add a border object widget, to place a border around a group of widgets.</p> <p>The widget is attached to the cursor. Click to drop the widget onto the tile.</p> <p>The widget can be filled and/or resized once it has been dropped onto the tile and its Border Mode property can select various forms of border.</p> |  |
|  Horizontal Spacer | <p>Click the Horizontal Spacer icon to add a horizontal spacer widget.</p> <p>Spacers are most relevant to the creation of Orbit MapView screens.</p> <p>Spacers are positioned between widgets to control on-screen layout. For example, to a preserved a layout when a set of widgets are scaled.</p> <p>They are used in Group Layouts.</p> |  |

Table 37 Widgets Tool Bar Items (Continued)



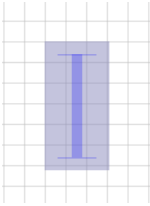
| Graphical Object | Description | On-Screen Example |
|--|--|---|
|  | | |
|  Vertical Spacer | <p>Click the Vertical Spacer icon to add a vertical spacer.</p> <p>Similar to Horizontal Spacer.</p> |  |

Table 37 Widgets Tool Bar Items (Continued)

C List of Behaviours and Bindings

List of Behaviours and Bindings

| | |
|---|-----|
| List of Behaviours | 333 |
| Alarm Behaviour (Group: Monitoring) | 334 |
| Alarm Mask Behaviour (Group: Monitoring) | 336 |
| Audio Level Behaviour (Group: Multiviewer) | 337 |
| Closed Caption Behaviour (Group: Multiviewer) | 337 |
| Display Details Behaviour (Group: Monitoring) | 339 |
| Local Value Behaviour (Group: General) | 339 |
| Log Field Behaviour (Group: Monitoring) | 340 |
| RollCall Timer Behaviour (Group: Timer) | 342 |
| RollCall v3 Command Behaviour (Group: RollCall) | 343 |
| RollCall+ Command Behaviour (Group: RollCall) | 345 |
| RollTrak Behaviour (Group: RollCall) | 345 |
| Schematic Link Behaviour (Group: General) | 347 |
| TRP Timer Behaviour (Group: Timer) | 348 |
| TSL Behaviour (Group: Multiviewer) | 349 |
| Video Input Behaviour (Group: Multiviewer) | 350 |
| List of Bindings | 352 |
| Audio Bars Binding (Group: Multiviewer) | 353 |
| Checkbox Binding (Group: General) | 355 |
| Direct Binding (Group: General) | 355 |
| Logical Binding (Group: General) | 356 |
| Mapped Value Binding (Group: General) | 360 |
| Property Binding (Group: General) | 363 |
| Radio Button Binding (Group: General) | 367 |
| RollCall Timer Binding (Group: Timer) | 368 |
| RollCall v3 Binding (Group: RollCall) | 369 |
| TRP Timer Binding (Group: Timer) | 369 |
| TSL UMD Binding (Group: Multiviewer) | 370 |

This appendix lists Orbit **Behaviours** and **Bindings** for use in multiviewer applications:

C.1 List of Behaviours

Behaviours can:

- Communicate with the outside world and perform some action.
- Receive events from widgets and process them, for example, sending a RollCall command.
- Carry one or more data values, for example, audio levels.

Behaviours are in the following groups:

General, Monitoring, Multiviewer, RollCall, and Timer.

The available **Behaviours** are listed in Table 38.

| Group | Behaviour Name | Link to (sub-section, page) | Applicable Widgets |
|--------------------|---------------------|-----------------------------|--|
| General | | | |
| | Local Value | (C.1.6, page 339) | All widgets |
| | Schematic Link | (C.1.12, page 347) | Button, Radio Button, Image |
| Monitoring | | | |
| | Alarm | (C.1.1, page 334) | Shapes, Tally, Image, Audio Bars, Video, Label |
| | Alarm Mask | (C.1.2, page 336) | Button, Radio Button, Image |
| | Display Details | (C.1.5, page 339) | Button, Radio Button, Image |
| | Log Field | (C.1.7, page 340) | Shapes, Tally, Video, Image, Label |
| Multiviewer | | | |
| | Audio Level | (C.1.3, page 337) | Audio Bars, Label |
| | Closed Caption | (C.1.4, page 337) | Label |
| | TSL | (C.1.14, page 349) | Rectangle Ellipse, Label, Image, Tally, UMD |
| | Video Input | (C.1.15, page 350) | Label, Audio, Video |
| RollCall | | | |
| | RollCall V3 Command | (C.1.9, page 343) | Shapes, UI Controls, Tally, Image, Label |
| | RollCall+ Command | (C.1.10, page 345) | Shapes, UI Controls, Tally, Image, Label |
| | RollTrak | (C.1.11, page 345) | Button, Radio Button, Image |
| Timer | | | |
| | RollCall Timer | (C.1.10, page 345) | Timer, Label |
| | TRP Timer | (C.1.13, page 348) | Timer, Label |

Table 38 Behaviours

The **Behaviours** are described in alphabetical order in the following sub-sections.

C.1.1 Alarm Behaviour (Group: Monitoring)

The Alarm **Behaviour** connects to device(s) or to log field header(s) and holds the aggregate state of all selected log field headers.

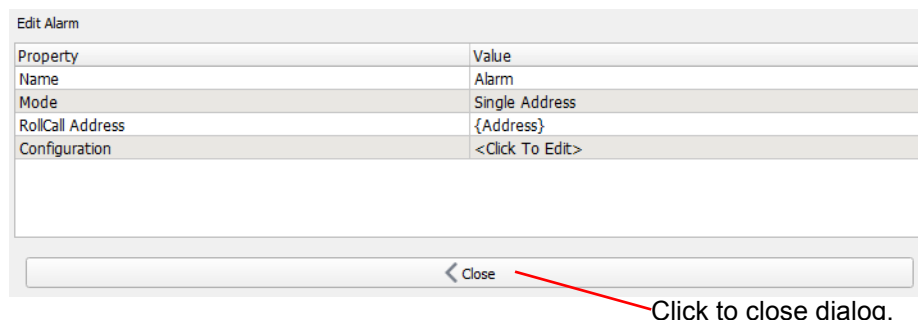


Figure 387 Behaviour Properties Dialog

| Property | Description |
|-----------------------------------|---|
| Name | Enter a unique name for the Behaviour block. |
| Mode | Drop-down box. |
| Single Address | Looks only at the unit specified by the RollCall Address property. (Used to get aggregate state of a unit without needing to select individual log field headers) |
| Multiple Addresses/Headers | Units/Log field headers are selected by the Configuration property. The RollCall Address field is not used. |
| RollCall Address | Enter RollCall address of the target device. |
| Configuration | Click to select all the Units and Log Headers of interest. The Unit and Header selection dialog is shown, see Figure 388. Select the required Headers from the required Unit(s)/Input(s). Only live information is shown. |

Table 39 Behaviour Properties

Behaviour Values:

| Item | Type | Description |
|--------------|--------|--|
| State | Enum | <ul style="list-style-type: none"> 1, Okay 49, Acknowledged Warning 50, Unacknowledged Warning 99, Acknowledged Error 100, Unacknowledged Error |
| Cause | String | Log field header value. Multiple alarms conditions are shown with a "+X", where X is the number of additional alarm states active |

Table 40 Behaviour Output Values, Available to a Binding

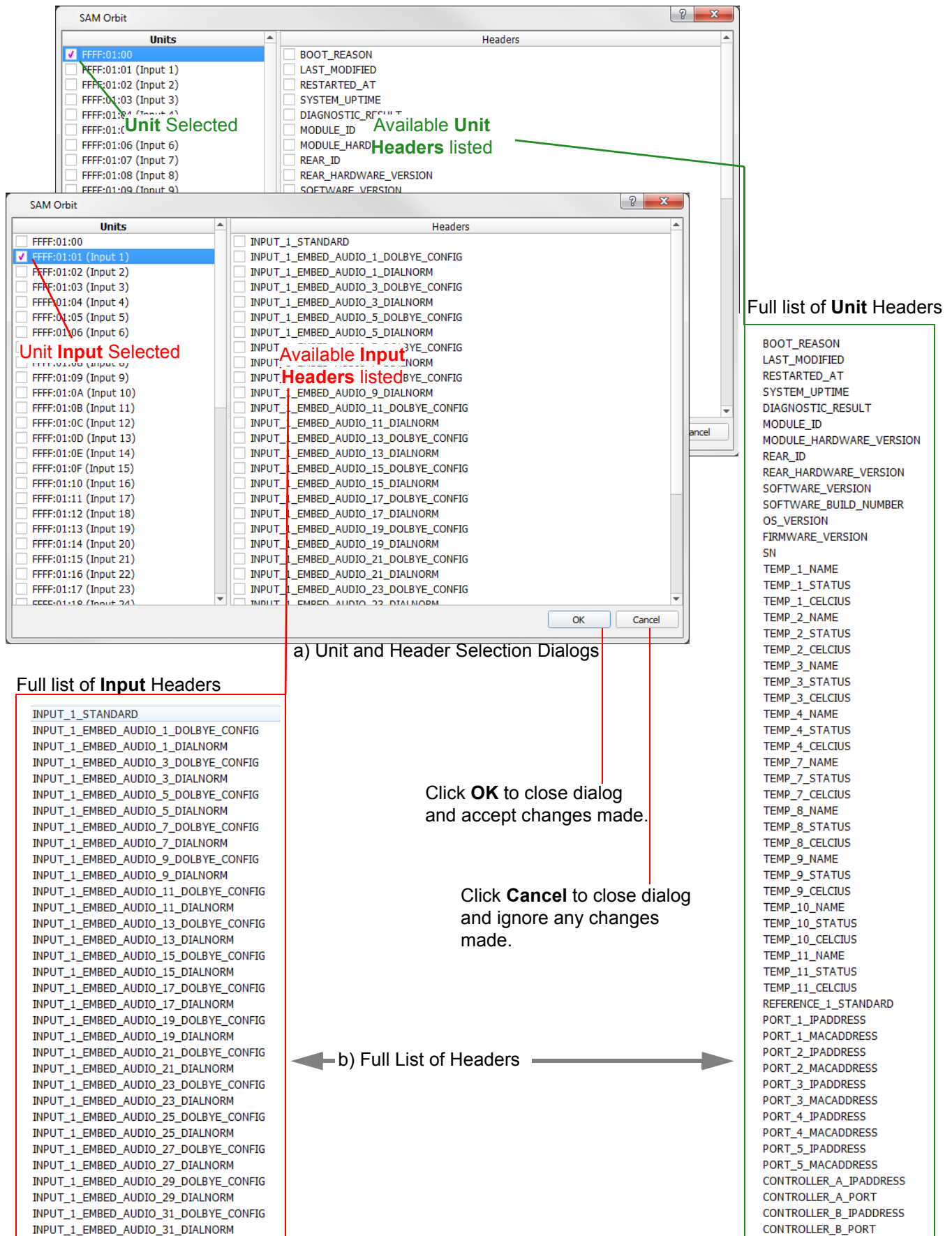


Figure 388 Configuration Property - Unit and Header Selection Dialog:

a) Unit and Headers Selection Dialog.

b) Full List of Headers.

C.1.2 Alarm Mask Behaviour (Group: Monitoring)

Mask or Unmask alarms from specific units and/or log field headers.

Click to close dialog.

Figure 389 Behaviour Properties Dialog

| Property | Description |
|----------------------|---|
| Name | Enter a unique name for the Behaviour block. |
| Configuration | Selection of specific devices and log headers. Click to select units and log headers in the Unit and Headers Selection dialog. (See Figure 388.) Data shown in the configuration screen is live data. |
| Mask Mode | Drop down box. <div> Mask Mask log fields set up in the Configuration property. Log field values will then be hidden (masked) from the system. </div> <div> Unmask Clear mask on log fields set up the in Configuration property. Log field values will be available to the system (i.e. unmasked). </div> |

Table 41 Behaviour Properties

| Item | Type | Description |
|--------------|------|--|
| Value | Enum | <ul style="list-style-type: none"> 0: Mask 1: Unmask |

Table 42 Behaviour Output Values, Available to a Binding

Click to jump to [Section C "List of Behaviours and Bindings"](#)

C.1.3 Audio Level Behaviour (Group: Multiviewer)

Carries audio level data from a source unit. Up to 32 audio levels are carried in the behaviour.

Click to close dialog.

Figure 390 Behaviour Properties Dialog

| Property | Description |
|-------------------------|--|
| Name | Enter a unique name for the Behaviour block. |
| RollCall Address | Enter the RollCall address of the device/unit of interest. |
| Input | Enter the input number on the device/unit of interest. |

Table 43 Behaviour Properties

| Item | Type | Description |
|-----------------|------------------|--|
| Channel 1 to 32 | Array of Integer | One array of one or more audio level values. One value per audio channel. Range 0 to 100. 0 indicates silence, 100 indicates full scale. |

Table 44 Behaviour Output Values, Available to a Binding

Click to jump to [Section C "List of Behaviours and Bindings"](#)

C.1.4 Closed Caption Behaviour (Group: Multiviewer)

The Closed Caption **Behaviour** carries closed caption data.

Figure 391 Behaviour Properties Dialog

| Property | Description |
|-------------------------|---|
| Name | Enter a unique name for the Behaviour block. |
| RollCall Address | Enter the RollCall address of the device/unit of interest. |
| Input | Enter the input number on the device/unit of interest. |
| Page | <p>Enter the closed caption page number of interest.</p> <p>On MV-8 Series multiviewers, 'closed captions' are supported for EIA-608 "line 21" and for EIA-708 in compatibility-byte mode. Page numbers:</p> <ul style="list-style-type: none"> • CC1 - page 101 • CC2 - page 102 • CC3 - page 103 • CC4 - page 104 • WST - page 888 (wide-screen signaling, aspect ratio) • OP42 - page 801 (typically) • OP47 - page 801 (typically) |

Note: For SD video input sources, the vertical blanking interval (VBI) is used:

For 625/25:

- VITC is detected on line 19;
- OP42/WST on line 21; and
- WSS on line 23.

For 525/29:

- VITC is detected on line 14; and
- Rec. 608 on line 21.

For HD/3G video input sources, the ancillary data (ANC) is used and there is no line number restriction.

Table 45 Behaviour Properties

| Item | Type | Description |
|-------|--------|--------------------------|
| Value | String | The closed caption text. |

Table 46 Behaviour Output Values, Available to a Binding

Click to jump to [Section C "List of Behaviours and Bindings"](#)

C.1.5 Display Details Behaviour (Group: Monitoring)

Displays the 'details window' for a specific device address.

| Property | Value |
|------------------|-----------------|
| Name | Display Details |
| RollCall Address | {Address} |

Close

Click to close dialog.

Figure 392 Behaviour Properties Dialog

| Property | Description |
|------------------|---|
| Name | Enter a unique name for the Behaviour block. |
| RollCall Address | A single RollCall address |

Table 47 Behaviour Properties

| Item | Type | Description |
|-------|--------|------------------------------|
| Value | String | The device RollCall address. |

Table 48 Behaviour Output Values, Available to a Binding

Click to jump to [Section C "List of Behaviours and Bindings"](#)

C.1.6 Local Value Behaviour (Group: General)

Used to hold an interim local value. This may be a value from a widget, for example, a slider value, or an interim value resulting from a logical operation, for example, a Logical Binding.

| Property | Value |
|----------|---------------|
| Name | Initial Value |

Close

Click to close dialog.

Figure 393 Behaviour Properties Dialog

| Property | Description |
|----------|---|
| Name | Enter a unique name for the Behaviour block. |

Table 49 Behaviour Properties

| Property | Description |
|----------------------|--|
| Initial Value | The initial value for the Behaviour to hold. When the Behaviour is exercised in 'Run Mode' or when it is running on a multiviewer, the Behaviour or Binding adopts values from the widget that it is linked to. |

Table 49 Behaviour Properties

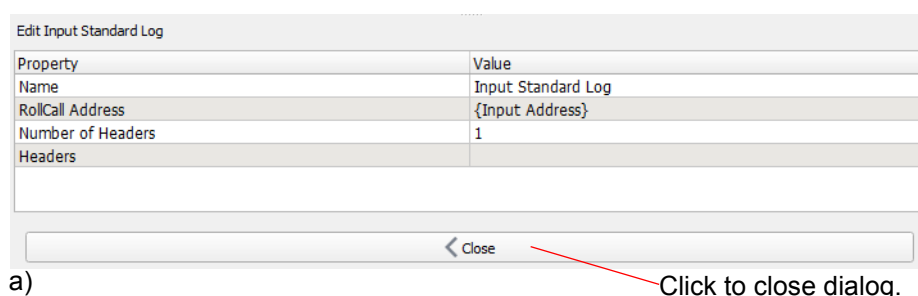
| Item | Type | Description |
|--------------|---|--|
| Value | Depends on the widget value it is connected to. | The value held by the Behaviour or Binding . |

Table 50 Behaviour Output Values, Available to a Binding

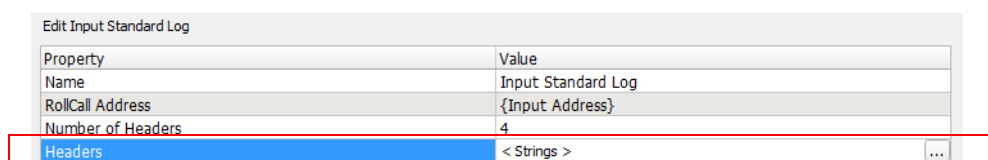
Click to jump to [Section C "List of Behaviours and Bindings"](#)

C.1.7 Log Field Behaviour (Group: Monitoring)

Connects to a device and reads the value of specified log fields.



a)



b)

Figure 394 Behaviour Properties Dialog:

a) Initial Dialog.

b) Headers Property Field.

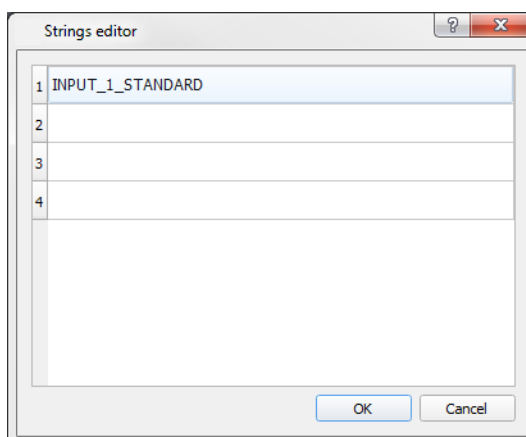
| Property | Description |
|--------------------------|--|
| Name | Enter a unique name for the Behaviour block. |
| RollCall Address | The RollCall address of device to read the log fields from. Can be a specific address or a variable defined in the project. |
| Number of Headers | The number of log fields to access, minimum 1, maximum 32. |

Table 51 Behaviour Properties

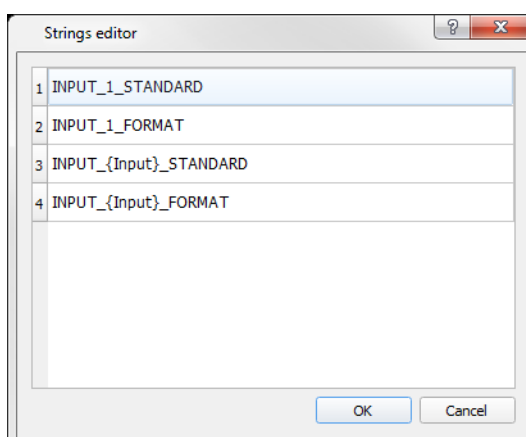
| Property | Description |
|----------------|--|
| Headers | <p>Click in the value field and click on ... (see Figure 394b). A Strings Editor dialog is shown, see Figure 395, showing a list of the log fields to access. (The number of list items is determined by the Number of Headers property value.)</p> <p>Enter the required log header field names. The names can contain variables (for example, INPUT_{Input}_FORMAT) which are evaluated at run time by Orbit.</p> <p>(The use of variables enables the functionality set up on a Tile/Schematic to be used generically throughout a Orbit project.)</p> |

Table 51 Behaviour Properties

| Item | Type | Description |
|---------------------------------------|-----------------|--|
| Title of Log Field defined in Headers | Array of String | Array of one or more log field values. One array item per log field configured |

Table 52 Behaviour Output Values, Available to a Binding

a)



b)

Figure 395 Strings Editor Dialog:
a) Initial empty dialog.
b) Filled out example.

Click to jump to [Section C "List of Behaviours and Bindings"](#)

C.1.8 RollCall Timer Behaviour (Group: Timer)

The MV-8 series multiviewers support up to 16 timers which are controlled through RollCall. The multiviewer makes the time data available through the **Timer Behaviour**.

| Property | Value |
|----------|----------------|
| Name | RollCall Timer |
| Timer | Timer 6 |

Close

Figure 396 Behaviour Properties Dialog

| Property | Description |
|--------------|--|
| Name | Enter a unique name for the Behaviour block. |
| Timer | Drop down box. See Figure 397. <ul style="list-style-type: none"> None: Ignore updates Timer 1: Time updates from Timer 1 Timer 2: Time updates from Timer 2 ... Timer 16: Time updates from Timer 16 |

Table 53 Behaviour Properties

| Item | Type | Description |
|-------------------|---------|---|
| Timer ID | Enum | <ul style="list-style-type: none"> 0, Invalid 1, Timer 1 2, Timer 2 ... 16, Timer 16 |
| Time | Integer | Time in milliseconds. |
| TimeString | String | (-)HH:MM:SS |
| Online | Boolean | <ul style="list-style-type: none"> True, Online False, Offline |
| Running | Boolean | <ul style="list-style-type: none"> True, Running False, Stopped |

Table 54 Behaviour Output Values, Available to a Binding

| Item | Type | Description |
|-----------|---------|---|
| Increment | Boolean | <ul style="list-style-type: none"> • True, Count Up • False, Count Down |

Table 54 Behaviour Output Values, Available to a Binding

| Timer | None |
|-------|----------|
| | Timer 1 |
| | Timer 2 |
| | Timer 3 |
| | Timer 4 |
| | Timer 5 |
| | Timer 6 |
| | Timer 7 |
| | Timer 8 |
| | Timer 9 |
| | Timer 10 |
| | Timer 11 |
| | Timer 12 |
| | Timer 13 |
| | Timer 14 |
| | Timer 15 |
| | Timer 16 |

Figure 397 Timer Property Values

Click to jump to [Section C "List of Behaviours and Bindings"](#)

C.1.9 RollCall v3 Command Behaviour (Group: RollCall)

The RollCall v3 **Behaviour** reads or writes the value of a RollCall v3 command.

| Property | Value |
|------------------|--------------------|
| Name | |
| Ip Index | Use Map Connection |
| RollCall Address | {Address} |
| Command Number | 0 |
| Write Mode | Read/Write |

Close

Figure 398 Behaviour Properties Dialog

| Property | Description |
|--|---|
| Name | Enter a unique name for the Behaviour block. |
| IP Index | Drop-down box. Select connection, see Figure 399. |
| Use Map Connection | Use a Network Map Connection, set up in the Orbit RollCall > Network Map Open Network dialog. |
| IP Share 1 ... IP Share 10 | Use one of the Direct IP Share Connections, set up in the Orbit RollCall > Network Map Open Network dialog. |
| RollCall Address | Enter the RollCall address of the target device. |
| Command Number | Enter the RollCall command number for the value to be read/written. For example, 1403 is the RollCall command to enable /disable GPIO3. |
| Write Mode | The write mode of the command being accessed, Read, Write or Read/Write |

Table 55 Behaviour Properties

| Item | Type | Description |
|-------|---|---|
| Value | Depends on widget value connected to, and the RollCall command specified. | The value to be read from or written to the RollCall command. |

Table 56 Behaviour Output Values, Available to a Binding

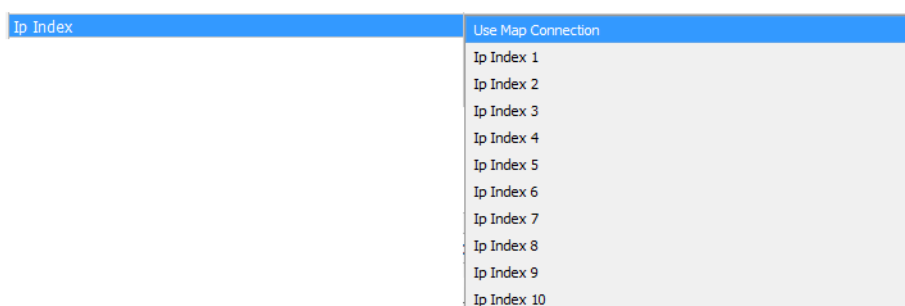


Figure 399 IP Index Property Values

Click to jump to [Section C "List of Behaviours and Bindings"](#)

C.1.10 RollCall+ Command Behaviour (Group: RollCall)

The RollCall+ Command **Behaviour** reads or writes the value of a RollCall+ command.

| Property | Value |
|------------------|-------------------|
| Name | RollCall+ Command |
| RollCall Address | {Address} |
| Command Number | 0 |
| Write Mode | Read/Write |

Close

Figure 400 Behaviour Properties Dialog

| Property | Description |
|-------------------------|--|
| Name | Enter a unique name for the Behaviour block. |
| RollCall Address | Enter the RollCall address of the target device. |
| Command Number | Enter the RollCall command number for the value to be read/written. |
| Write Mode | Select the write mode of the command being accessed, Read, Write or Read/Write |

Table 57 Behaviour Properties

| Item | Type | Description |
|-------|---|---|
| Value | Depends on widget value connected to, and on the RollCall+ command specified. | The value to be read from or written to the RollCall command. |

Table 58 Behaviour Output Values, Available to a Binding

Click to jump to [Section C "List of Behaviours and Bindings"](#)

C.1.11 RollTrak Behaviour (Group: RollCall)

The RollTrak **Behaviour** sets or gets RollTrak information using the RollCall v3 protocol.

| Property | Value |
|---------------------|--------------------|
| Name | |
| IP Index | Use Map Connection |
| RollCall Address | {Address} |
| Command Number | 0 |
| RollTrak Type | Get |
| RollTrak Value Type | Int |
| String Value | |
| Int Value | 0 |

Figure 401 Behaviour Properties Dialog

| Property | Description |
|--|---|
| Name | Enter a unique name for the Behaviour block. |
| IP Index | Drop-down box. Select connection, see Figure 399. |
| Use Map Connection | Use a Network Map Connection, set up in the Orbit RollCall > Network Map Open Network dialog. |
| IP Share 1 ... IP Share 10 | Use one of the Direct IP Share Connections, set up in the Orbit RollCall > Network Map Open Network dialog. |
| RollCall Address | The RollCall address of the target unit. |
| Command Number | The RollCall command number for the value to be read/written. |
| RollTrak Type | <ul style="list-style-type: none"> Get: Read Value. Set: Write Value. |
| RollTrak Value Type | <ul style="list-style-type: none"> Int: Integer type. String: String type. Preset: Set command back to preset value. |
| String Value | String to use with 'String' RollTrak value type. |
| Int Value | Integer to use with 'Int' RollTrak value type. |

Table 59 Behaviour Properties

| Item | Type | Description |
|-------|-------------------------|--|
| Value | String or Integer | The value returned from a 'Get' operation. |

Table 60 Behaviour Output Values, Available to a Binding

C.1.12 Schematic Link Behaviour (Group: General)

This is a link between two Orbit **Schematics**. It can provide navigation through an Orbit MapView project.

| Property | Value |
|---------------------|--|
| Name | SchematicLink |
| Single Click | <input checked="" type="checkbox"/> True |
| Path | /tiles/My-Sch-Link-2.tile |
| Link Mode | Schematic Link |
| Open in Same Window | <input checked="" type="checkbox"/> True |

Close

Figure 402 Behaviour Properties Dialog

| Property | Description |
|----------------------------|--|
| Name | Enter a unique name for the Behaviour block. |
| Single Click | Check box. True: Single-Click to follow link. False: Double-Click to follow link. |
| Path | Enter a link to the Orbit Schematic , Tile or Panel file to be opened by the link. Either: <ul style="list-style-type: none"> Click in the value field. Enter the path, for example, /tiles/My-Tile.tile or <ul style="list-style-type: none"> Click in the value field. Click the Browse Path Icon, see Figure 403a. Select the path via the Select File dialog, see Figure 403b. |
| Link Mode | Drop-down box. |
| Schematic Link | Opens a specific Orbit file. |
| Back Button | Opens an earlier Orbit file in a history of open Orbit files. |
| Forward Button | Opens a later Orbit file in a history of open Orbit files. |
| Open in Same Window | Check box. True: Opens file in the same tab. False: Opens file in a new tab. |

Table 61 Behaviour Properties

| Item | Type | Description |
|------|------|-------------|
| None | - | - |

Table 62 Behaviour Output Values, Available to a Binding

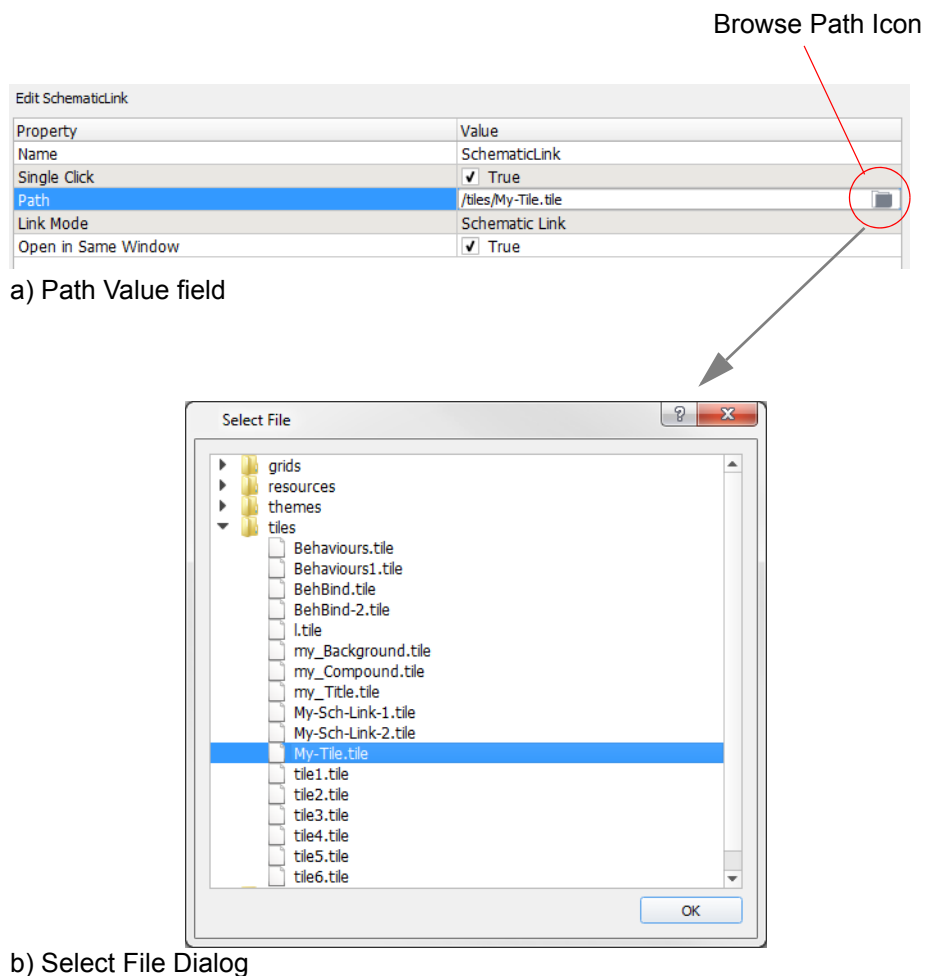


Figure 403 Selecting the Path Property Value

Click to jump to [Section C "List of Behaviours and Bindings"](#)

C.1.13 TRP Timer Behaviour (Group: Timer)

The Plura[®]Timer Request Protocol is originated from a Plura[®] timer controller and is decoded by a MV-8 series multiviewer device.

The timer data is made available through the TRP Timer **Behaviour**. The **Behaviour**'s value is updated with timer data.

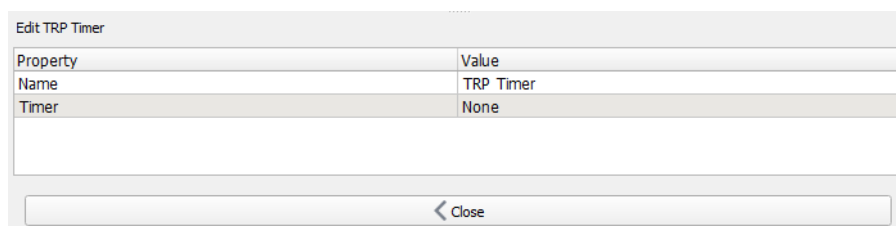


Figure 404 Behaviour Properties Dialog

| Property | Description |
|-----------------|---|
| Name | Enter a unique name for the Behaviour block. |
| Timer | Drop down box. Select which timer data to use to update the Behaviour value. |
| None | Ignore all timer data updates. |
| Timer A | Use Timer A, |
| Timer B | Timer B, |
| ... | ... |
| Timer F | Timer F. |
| Time | Use time of day. |
| Date | Use date. |
| TimeCode | Use timecode. |

Table 63 Behaviour Properties

| Item | Type | Description |
|---------------|--------|--|
| Timer ID | Enum | <ul style="list-style-type: none"> 0, Invalid 1, Timer A 2, Timer B ... 6, Timer F 7, Time 8, Date 9, TimeCode |
| Time | String | Time or date. |
| Status | Enum | <ul style="list-style-type: none"> 0, Flashing 1, On 2, Off |
| Color | Enum | <ul style="list-style-type: none"> 0, Red 1, Green 2, Yellow |

Table 64 Behaviour Output Values, Available to a Binding

C.1.14 TSL Behaviour (Group: Multiviewer)

The MV-8 series multiviewers decode the TSL UMD protocol.
The **Behaviour** makes the TSL data available in the Orbit project.

| Property | Value |
|----------------------|-------------|
| Name | TSL |
| TSL Version | Version 3.1 |
| 5.0: Screen Index | 0 |
| 5.0: Display Index | 0 |
| 3.1: Display Address | 0 |

Close

Figure 405 Behaviour Properties Dialog

| Property | Description |
|-----------------------------|---|
| Name | Enter a unique name for the Behaviour block. |
| TSL Version | Drop-down box. Select the version of the TSL protocol. |
| Version 3.1 | Version 3.1 |
| Version 5.0 | Version 5.0 |
| 5.0: Screen Index | Integer: 0 to 65534. Normally an ID for a monitor. (Enabled when TSL Version is 'Version 5.0'.) |
| 5.0: Display Index | Integer: 0 to 65534. Normally an ID for a UMD on a monitor. (Enabled when TSL Version is 'Version 5.0'.) |
| 3.1: Display Address | Integer: 0 to 126. An ID for a UMD on a monitor. (Enabled when TSL Version is 'Version 3.1'.) |

Table 65 Behaviour Properties

| Item | Type | Description |
|---------|---------|------------------------|
| Tally 1 | Boolean | True, On False, Off |
| Tally 2 | Boolean | True, On False, Off |
| Tally 3 | Boolean | True, On False, Off |
| Tally 4 | Boolean | True, On False, Off |
| Text | String | The tally text. |

Table 66 TSL v3.1 Behaviour Output Values, Available to a Binding

| Item | Type | Description |
|------------|--------|-----------------|
| Left Lamp | Enum | 0 to 3 |
| Right Lamp | Enum | 0 to 3 |
| Text Tally | Enum | 0 to 3 |
| Label | String | The tally text. |

Table 67 TSL v5.0 Behaviour Output Values, Available to a Binding

C.1.15 Video Input Behaviour (Group: Multiviewer)

The Video Input **Behaviour** provides a source tally name for a video input.

| Property | Value |
|------------------|-------------|
| Name | Video Input |
| RollCall Address | {Address} |
| Input | {Input} |

Close

Figure 406 Behaviour Properties Dialog

| Property | Description |
|-------------------------|--|
| Name | Enter a unique name for the Behaviour block. |
| RollCall Address | Enter the RollCall address of the target device. The internal system variable Address may be used in the value field. For example, {Address} |
| Input | Enter the input number of the target device. The internal system variable Input may be used in the value field. For example, {Input} |

Table 68 Behaviour Properties

| Item | Type | Description |
|-------|--------|--------------------|
| Value | String | Source tally name. |

Table 69 Behaviour Output Values, Available to a Binding

Click to jump to [Section C "List of Behaviours and Bindings"](#)

C.2 List of Bindings

Bindings tie **Behaviour** values to widgets or to other **Behaviours**.

For example,

Displaying device status on a video wall tile:

- A device sends its status using RollCall messages.
- A **Behaviour** reads the RollCall messages.
- A **Binding** is used to synchronize the **Behaviour**'s value to an on-screen widget (to affect the widget's appearance).

All **Bindings** can be given a unique name.

Bindings have the following groups: General, Multiviewer, RollCall, and Timer.

The available **Bindings** are listed in Table 70.

| Group | Binding Name | (sub-section, page) | Applicable Widgets |
|--------------------|-----------------------|---------------------|---|
| General | | | |
| | Checkbox | (C.2.2, page 355) | Check box. |
| | Direct | (C.2.3, page 355) | All widgets. (Applies to a Tile or a Schematic .) |
| | Logical | (C.2.4, page 356) | All widgets. (Applies to a Tile or a Schematic .) |
| | Mapped Value | (C.2.5, page 360) | All widgets. (Applies to a Tile or a Schematic .) |
| | Property | (C.1.6, page 363) | All widgets. |
| | Radio Button | (C.1.6, page 339) | Radio button. |
| Multiviewer | | | |
| | Audio Bars | (C.2.1, page 353) | Audio |
| | TSL UMD | (C.2.11, page 370) | UMD |
| RollCall | | | |
| | RollCall v3 | (C.2.9, page 369) | All widgets. (Applies to a Tile or a Schematic .) |
| Timer | | | |
| | RollCall Timer | (C.2.8, page 368) | Timer. |
| | TRP Timer | (C.1.6, page 369) | Timer. |

Table 70 Bindings

The **Bindings** are described in alphabetical order in the following sub-sections.

Click to jump to [Section C "List of Behaviours and Bindings"](#)

C.2.1 Audio Bars Binding (Group: Multiviewer)

Maps the level value of an audio channel from an Audio Levels **Behaviour** onto a specific audio level bar in an **Audio Bars** widget. The **Binding** properties automatically adjust to match the number of audio bars defined in the **Audio Bars** widget.

The **Binding** properties will show options for each bar specified in the **Audio Bars** widget. An **Audio Bars** widget can show up to 32 bars.

And the Audio Levels **Behaviour** can carry up to 32 levels.

A mapping is created between the audio level values and the audio bars.

The **Binding** is configured for different widget modes (PCM, Dolby E or Auto).

a) Audio Bars **Binding** Properties dialog, showing PCM Tab

b) Dolby E Tab

c) Auto Tab

Figure 407 Binding Properties Dialog:

a) Showing PCM Tab.

b) Dolby E Tab.

c) Auto Tab.

| Property | Description |
|--|--|
| Common Properties: | |
| Name | Enter a unique name for the Binding block. |
| Source Behaviour | <p>Select the Behaviour block for the Binding to interact with. This may already be automatically filled out.</p> <p>Click Select for a list of the names of available Behaviour blocks.</p> |
| PCM Tab: | |
| Bar 1: | |
| Channel | <p>Drop-down box. From the Behaviour's data value (Audio Levels Behaviour), select the audio channel (1 to 32).</p> <p>This audio level data will be shown on the corresponding audio bar.</p> |
| Bar 2: etc | There is an item for each audio bar in the Audio Bars widget. |
| Dolby E Tab: | |
| Bar 1: | |
| Pair | <p>Drop-down box. From the Behaviour's data value (Audio Levels Behaviour), select the audio channel stereo pair containing the Dolby E data to use.</p> <p>(Channel stereo pairs: 1-2, 3-4, ... or 31-32)</p> |
| Channel | <p>Drop-down box. Select the audio channel (1 to 8) from the selected Dolby E channel pair data.</p> <p>This audio level data will be shown on the corresponding audio bar.</p> <p>See Note 1.</p> |
| Bar 2: etc | There is an item for each audio bar in the Audio Bars widget |
| Auto Tab: | |
| Bar 1: | |
| Audio Level | <p>Drop-down box. Select Behaviour data value (1 to 32) to show on audio bar.(Audio Levels Behaviour),</p> <p>This audio level data will be shown on the corresponding audio bar.</p> <p>See Note 2.</p> |
| Bar 2: etc | There is an item for each audio bar in the Audio Bars widget. |
| <p>Note 1: Dolby E can contain up to 8 (compressed) audio channels per stereo channel-pair. In the first four stereo channel-pairs, there are up to 32 audio channels.</p> <p>An Audio Levels Behaviour is limited to Dolby E data in the first four stereo pairs. (Channel pairs 1-2, 3-4, 5-6 and 7-8).</p> <p>Note 2: The Audio Levels Behaviour can only carry up to 32 audio levels.</p> <p>If there is a mixture of PCM and Dolby E data present, only the first 32 audio levels can be carried by the Audio Levels Behaviour; it is not possible to show the audio levels from all 32 source channels.</p> | |

Table 71 Binding Properties

C.2.2 Checkbox Binding (Group: General)

Used to set the checked/unchecked state on a Checkbox widget.

Figure 408 Binding Properties Dialog

| Property | Description |
|-------------------------|--|
| Common Properties: | |
| Name | Enter a unique name for the Binding block. |
| Source Behaviour | <p>Select the Behaviour block for the Binding to interact with. This may already be automatically filled out.</p> <p>Click Select for a list of the names of available Behaviour blocks.</p> |
| Checked Value | Enter the Behaviour value to cause the Checkbox widget to be checked (selected). |
| Unchecked Value | Enter the Behaviour value to cause the Checkbox widget to be unchecked (unselected). |

Table 72 Binding Properties

Click to jump to [Section C "List of Behaviours and Bindings"](#)

C.2.3 Direct Binding (Group: General)

This is a simple **Binding** where both a **Behaviour** data value and a widget property data value are kept in synchronisation.

Normally a Direct **Binding** is created automatically by the Orbit GUI.

For example,

adding a Local Value **Behaviour** to a slider widget automatically creates a Direct **Binding**. The value in the widget (the slider position) and the value on the **Behaviour** are then kept in sync automatically.

Figure 409 Binding Properties Dialog

| Property | Description |
|-------------------------|---|
| Common Properties: | |
| Name | Enter a unique name for the Binding block. |
| Source Behaviour | <p>Select the Behaviour block for the Binding to interact with. This may already be automatically filled out.</p> <p>Click Select for a list of the names of available Behaviour blocks.</p> |
| Mode | Drop-down box: |
| Read | Only allowed to read the Behaviour value. |
| Write | Only allowed to write the Behaviour value. |
| Read/Write | Allowed to read and write the Behaviour value. |
| Format String | <p>Use C-style string formatting.</p> <p>Example, prefix a value with a string: Alarm State %s</p> <p>The %s will be replaced by the string value of interest.</p> <p>Example, show three digit number with leading zeros,: %03d</p> <p>A value of 99 will be shown as "099".</p> |

Table 73 Binding Properties

Click to jump to [Section C "List of Behaviours and Bindings"](#)

C.2.4 Logical Binding (Group: General)

This **Binding** is used to make a decision based on one or more **Behaviour** values: Several operations can be carried out on one or more **Behaviour** values and the results can be combined. The final result is either "true" or "false".

Literal or **Behaviour** values can then be mapped to these two interim results to form the **Binding**'s result, which is passed to either a target widget or to a **Behaviour**.

Operator Rule Table

Figure 410 Binding Properties Dialog

| Property | Description |
|-------------------------|---|
| Common Properties: | |
| Name | Enter a unique name for the Binding block. |
| Logical Operator | Drop-down box: |
| AND | The result of each Operator Rule Panel row is AND'ed together. |
| OR | The result of each Operator Rule Panel row is OR'ed together. |
| Operator Rule Table: | |
| | Rule rows are added for evaluation to form a list of rules. All the evaluated results are combined together according to the Logical Operator . |
| LHS Type: | Drop-down box: |
| Literal | LHS value is a literal value. |
| Behaviour | LHS value is a Behaviour value. |
| LHS Behaviour | Select the Behaviour for the LHS value. (Control is active when LHS Type is set to " Behaviour ") |
| LHS Value | LHS Type is Literal : Enter value required directly. For example "49". LHS Type is Behaviour : Select Behaviour value to use. |
| Operator | Drop-down box: Select operator. |
| | <div> Operator <div> Equals Not Equals Less Than Greater Than Less Than or Equal Greater Than or Equal Contains Text </div> </div> |
| RHS Type: | Drop-down box: |

Table 74 Binding Properties

| Property | Description |
|----------------------|---|
| Literal | LHS value is a literal value. |
| Behaviour | LHS value is a Behaviour value. |
| RHS Behaviour | Select the Behaviour for the LHS value. (Control is only active when LHS Type = " Behaviour ") |
| RHS Value | LHS Type = Literal : Enter value required directly. For example "49". LHS Type = Behaviour : Select Behaviour value to use. |
| Add | Button. Click to add a blank, new rule row into the rules table in the Operator Rule Table . |

| | |
|---------------|---|
| Delete | Button: Click to delete the selected row from the table of rules. |
|---------------|---|

Result: Map the "true"/"false" result to values.

True Result Drop-down box:

No Op No value is output from the **Binding**.

Literal Enter a literal, fixed value to use for the **Binding** value.

Behaviour Select a **Behaviour** to use for the Binding value.

- If the selected **Behaviour** has one value.

- If the Selected Behaviour has more than one value, then select the value to use with the drop-down box.

False Result Drop-down box. (Similar to **True Result** drop-down box.)

Table 74 Binding Properties

| Property | Description |
|------------------|--|
| Target | Radio Button: Determines where to send the Binding value. |
| Widget | Binding value is connected to the widget on the graphical Behaviour and Binding Editor screen. |
| Behaviour | Binding value is connected to a selected Behaviour . |
| Select | Button. Click to select the Behaviour . |

Table 74 Binding Properties

See Figure 411 for an example Logical **Binding** block configuration.

Example:

The example operates on two status values, "MY Status" and "MY Status 2".

(Each status value is in the range 0 to 100:

- 0: "Status OK",
- 49: "Acknowledged Warning",
- 50: "Warning",
- 99: "Acknowledged Error",
- 100: "Error".)

The Logical **Binding** block "Combine Status" combines the two status values.

The block passes on the higher status value to be stored a Local Value **Behaviour** (Combined Status). The Combined Status value is mapped in a Property Binding (Status-to-Tally-Color) to be suitable to control the Tally widget's lamp color.

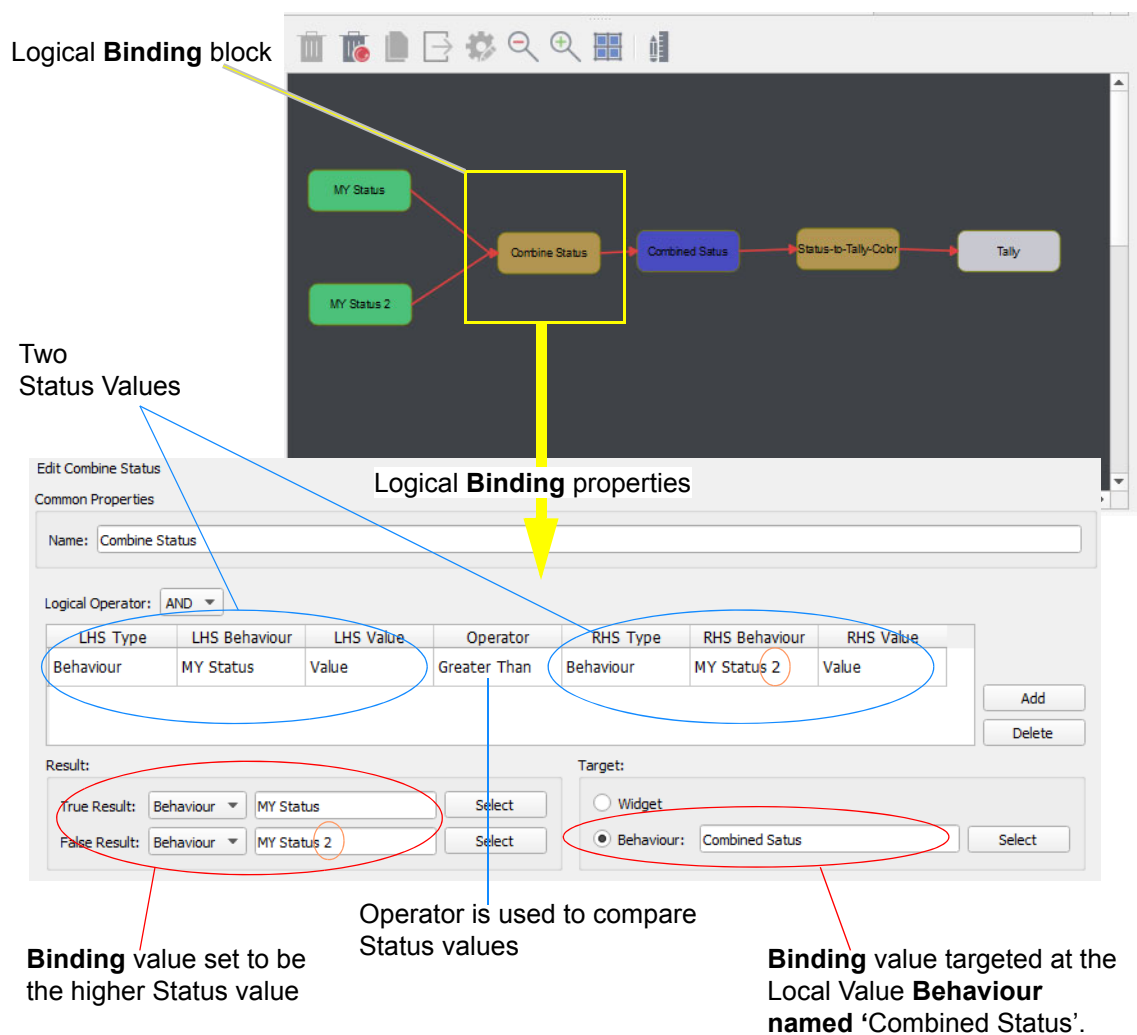


Figure 411 Logical Binding Example

Click to jump to [Section C "List of Behaviours and Bindings"](#)

C.2.5 Mapped Value Binding (Group: General)

This **Binding** applies simple logic to a specific value from a **Behaviour**, performing simple logic rules.

The result is sent to either the widget it is bound to or to another **Behaviour**'s value.

Common Properties

Name:

Source Behaviour:

Property to bind:

| Bind Rules: | Operator | Expression | Result |
|-------------|----------|------------|--------|
| | | | |

Target: ☒ Widget ☐ Behaviour

Bind Rules Table

Figure 412 Binding Properties Dialog

| Property | Description |
|-------------------------|---|
| Common Properties: | |
| Name | Enter a unique name for the Binding block. |
| Source Behaviour | <p>Select the Behaviour block for the Binding to interact with. This may already be automatically filled out.</p> <p>Click Select for a list of the names of available Behaviour blocks.</p> <p>If required, further select which of the Behaviour's values to use.</p> |
| Property to bind: | |
| Bind Rules | <p>Processing rules are listed in the Bind Rules Table.</p> <p>Note: The first rule to be satisfied is used. Thus, the ordering of the rules in the table is critical.</p> |
| Operator | <p>Drop-down box:</p> <p>Select operator.</p> <div> <div>Operator</div> <div> Equals Not Equals Less Than Greater Than Less Than or Equal Greater Than or Equal Contains Text </div> </div> |
| Expression | <p>Enter literal, fixed value.</p> <p>This value is compared with the Behaviour value. For example, "Greater Than or Equal" to 49.</p> |

Table 75 Binding Properties

| Property | Description |
|------------------|---|
| Result | Enter value to use if expression evaluates to true. This can be a literal value, a color or a string. For example, 49, or #00FF00, or "Warning" Note: In a string value type, "%1" can be used to represent the Behaviour value used. If the Behaviour has more than one value, then %2, %3 etc can also be used. |
| Move Down | Button. Click to move the selected row down the list. |
| Move Up | Button. Click to move the selected row up the list. |
| Add | Button. Click to add a new, blank rule row to the list. |
| Delete | Button. Click to delete the selected row from the list. |
| Target | Radio Button: Determines where to send the Binding value. |
| Widget | Binding value is connected to the widget shown on the graphical Behaviour and Binding Editor screen. |
| Behaviour | Binding value is connected to a selected Behaviour . |
| Select | Button. Click to select the Behaviour . |

Table 75 Binding Properties (Continued)

See Figure 413 for an example Mapped Value **Binding** block configuration.

The example operates on a status value, “Combined Status”.

(Each status value is in the range 0 to 100:

- 0: “Status OK”,
- 49: “Acknowledged Warning”,
- 50: “Warning”,
- 99: “Acknowledged Error”,
- 100: “Error”.)

The Mapped Value **Binding** block processes the status value to yield a string value suitable for display in a Label widget. The processing rules are listed in the Bind Rules Table.

The block passes on the higher status value to be stored a Local Value **Behaviour** (Combined Status). The Combined Status value is mapped in a Property Binding (Status-to-Tally-Color) to be suitable to control the Tally widget’s lamp color.

“Combined Status” **Behaviour** source value Label widget block

Mapped Value
Binding Block

The diagram illustrates the configuration of a Mapped Value Binding block. The top part shows a workflow with three blocks: 'Combined Status' (green), 'Mapped Value' (yellow), and 'Label' (grey). Arrows indicate the flow from 'Combined Status' to 'Mapped Value' and then to 'Label'. A yellow arrow points from the 'Mapped Value' block to the 'Binding Properties' window below.

The 'Binding Properties' window is titled 'Edit Mapped Value' and contains the following sections:

- Common Properties:**
 - Name: Mapped Value
 - Source Behaviour: Combined Status (with a 'Select' button)
- Property to bind:**
 - Bind Rules: A table with columns 'Operator', 'Expression', and 'Result'.

| Operator | Expression | Result |
|-----------------------|------------|--------------|
| Greater Than or Equal | 99 | ERROR (%1) |
| Greater Than or Equal | 49 | WARNING (%1) |
| Default | | OK (%1) |
- Target:**
 - ☒ Widget (circled in red)
 - ☐ Behaviour

At the bottom right of the window are buttons: 'Move Down', 'Move Up', 'Add', and 'Delete'. A 'Select' button is also present at the bottom right.

Binding value targeted at widget.

Figure 413 Mapped Value Binding Example

Click to jump to [Section C "List of Behaviours and Bindings"](#)

C.2.6 Property Binding (Group: General)

The Property **Binding** is used to change a property on a widget from a specific value in a **Behaviour**.

Performs simple logic rules on the behaviours value and applies the result to a specific property on the widget.

Bind Rules Table

Figure 414 Binding Properties Dialog

| Property | Description |
|--------------------------|---|
| Common Properties: | |
| Name | Enter a unique name for the Binding block. |
| Source Behaviour | <p>Select the Behaviour block for the Binding to interact with. This may already be automatically filled out.</p> <p>Click Select for a list of the names of available Behaviour blocks.</p> <p>If required, further select which of the Behaviour's values to use.</p> |
| Property to bind: | <p>Drop-down box.</p> <p>Select which property of the widget to bind to.</p> <p>Note: The drop-down list items shown depends on the widget being bound to.</p> |
| Bind Rules | <p>Processing rules are listed in the Bind Rules Table.</p> <p>The first rule to be satisfied is used. The ordering of the rules in the list is critical.</p> |
| Operator | <p>Drop-down box.</p> <p>Select operator.</p> |

Table 76 Binding Properties

| Property | Description |
|-------------------|---|
| Expression | Enter literal, fixed value. This value is compared with the Behaviour value. For example, 49, or "Warn" |
| Result | Enter value to use if expression evaluates to true. This can be a literal value, a color or a string. For example, 49, or #00FF00, or "Warning" Note: In a string value type, "%1" can be used to represent the Behaviour value used. If the Behaviour has more than one value, then %2, %3 etc can also be used. |
| Move Down | Button. Click to move the selected row down the list. |
| Move Up | Button. Click to move the selected row up the list. |
| Add | Button. Click to add a new, blank rule row to the list. |
| Delete | Button. Click to delete the selected row from the list. |
| Target | Radio Button: Determines where to send the Binding value. |
| Widget | Binding value is connected to the widget shown on the graphical Behaviour and Binding Editor screen. |
| Behaviour | Binding value is connected to a selected Behaviour . |
| Select | Button. Click to select the Behaviour . |

Table 76 Binding Properties

See Figure 416 for an example Property **Binding** block configuration.

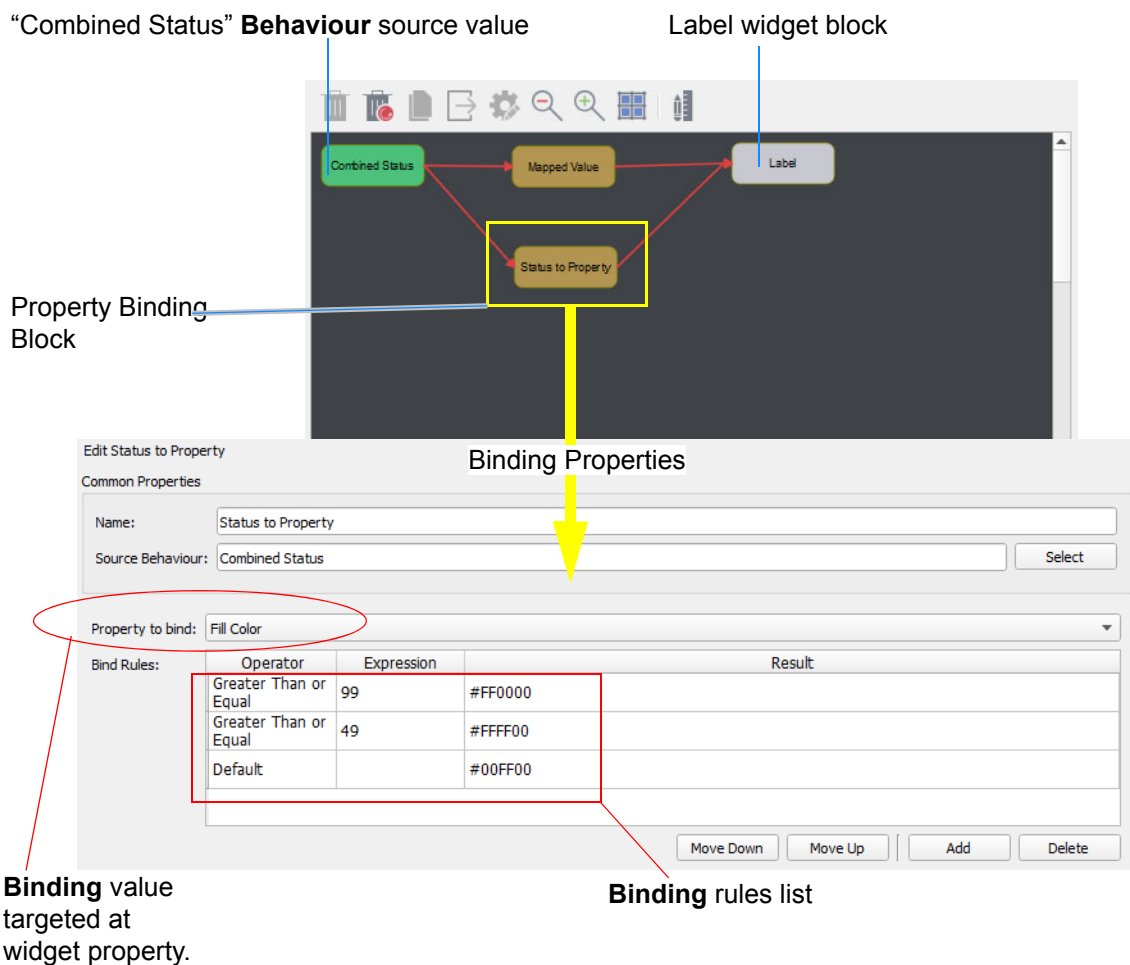
The example operates on a status value, “Combined Status”.

(Each status value is in the range 0 to 100:

- 0: “Status OK”,
- 49: “Acknowledged warning”,
- 50: “Warning”,
- 99: “Acknowledged Error”,
- 100: “Error”.)

The Mapped Value **Binding** block processes the status value to yield a value suitable for binding to the Fill Color property of a Label widget. The processing rules are listed in the Bind Rules Panel.

The block passes on the resulting value to the widget’s property, controlling the Label’s fill color according to the status value.



Property to Bind drop-down list.

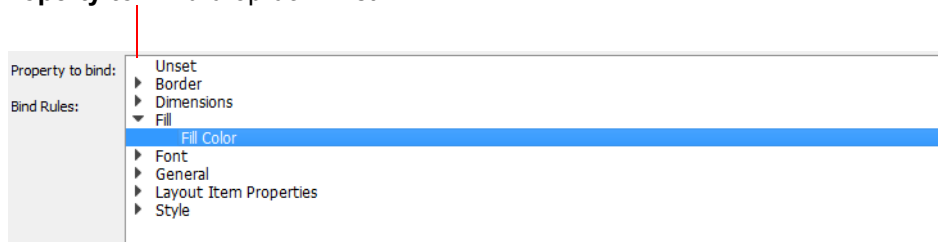


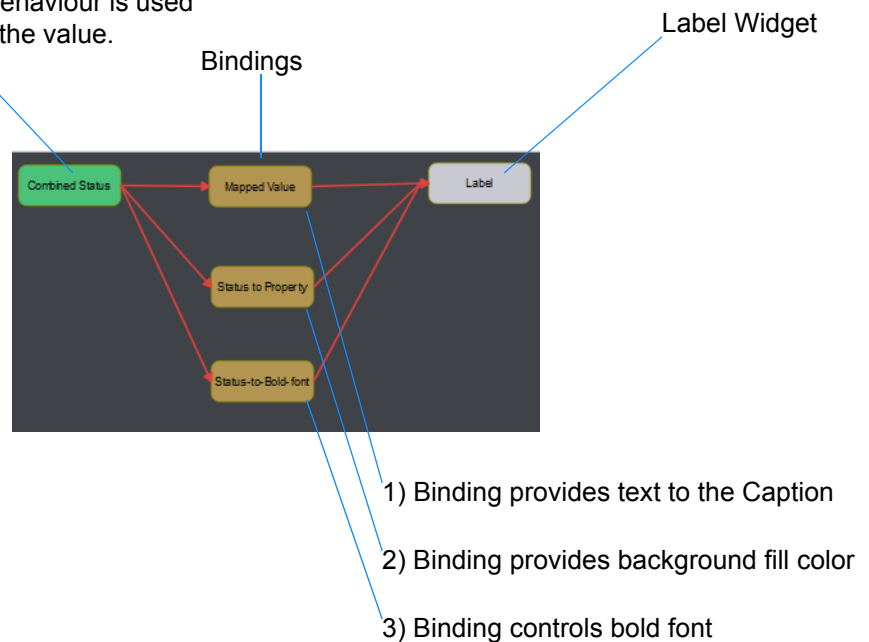
Figure 415 Property Binding Example

More than one **Binding** may be bound to a widget; Figure 416 shows this.

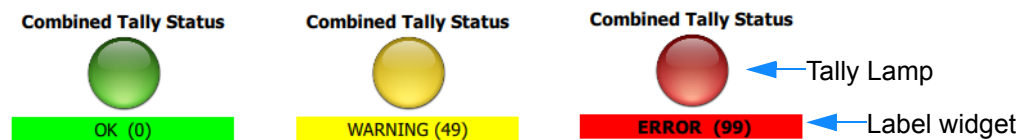
An extra Property **Binding** has been added, to control the widget's Label text bold font. The graphical **Behaviours and Bindings Editor** shows the structure of the **Binding(s)** and **Behaviour(s)** for a widget.

Behaviour:

A Local Value Behaviour is used as a source for the value.



a) **Behaviours and Bindings** controlling a Label widget.



b) On-screen appearance of Label Widget (A Tally widget is also shown)

Figure 416 Property Binding Example:

- a) **Behaviours and Bindings Controlling a Label Widget.**
- b) **On-screen Appearance of the Label Widget**
(Also shown is a Lamp Tally Widget).

Click to jump to [Section C "List of Behaviours and Bindings"](#)

C.2.7 Radio Button Binding (Group: General)

The Radio Button **Binding** is used to set the “selected” state of a Radio Button widget. This is typically used with RollCall V3 Command or RollCall+ Command **Behaviours**.

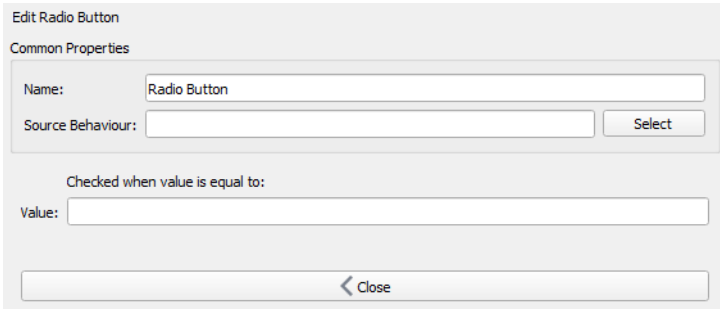


Figure 417 Binding Properties Dialog

| Property | Description |
|------------------------------|---|
| Common Properties: | |
| Name | Enter a unique name for the Binding block. |
| Source Behaviour | <p>Select the Behaviour block for the Binding to interact with. This may already be automatically filled out.</p> <p>Click Select for a list of the names of available Behaviour blocks.</p> <p>If required, further select which of the Behaviour’s values to use.</p> |
| Checked when Value Equal to: | |
| Value | <p>Enter the value of the Source Behaviour Value for the Radio Button to be shown as “selected”.</p> <p>The Radio Button can reflect the value of the source value or, if the Radio Button widget is manually checked on-screen, it can set the source value.</p> |

Table 77 Binding Properties

Click to jump to [Section C "List of Behaviours and Bindings"](#)

C.2.8 RollCall Timer Binding (Group: Timer)

The RollCall Timer **Binding** is used to bind data from the RollCall Timer **Behaviour** to a Timer widget. There are no user configuration items.

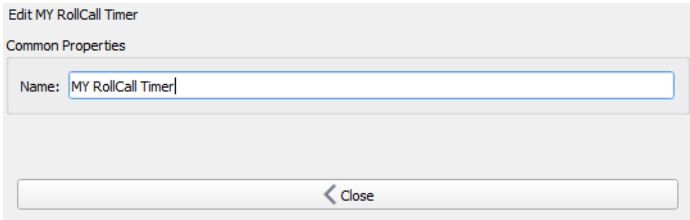


Figure 418 Binding Properties Dialog

| Property | Description |
|--------------------|---|
| Common Properties: | |
| Name | Enter a unique name for the Binding block. |

Table 78Binding Properties

C.2.9 RollCall v3 Binding (Group: RollCall)

The RollCall V3 **Binding** is used to bind data from the RollCall V3 **Behaviour**. There are no user configuration items.

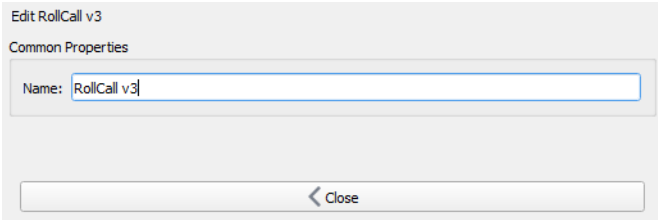


Figure 419 Binding Properties Dialog

| Property | Description |
|--------------------|---|
| Common Properties: | |
| Name | Enter a unique name for the Binding block. |

Table 79Binding Properties

Click to jump to [Section C "List of Behaviours and Bindings"](#)

C.2.10 TRP Timer Binding (Group: Timer)

The TRP Timer **Binding** is used to bind data from the TRP Timer **Behaviour** to a Timer widget. There are no user configuration items.

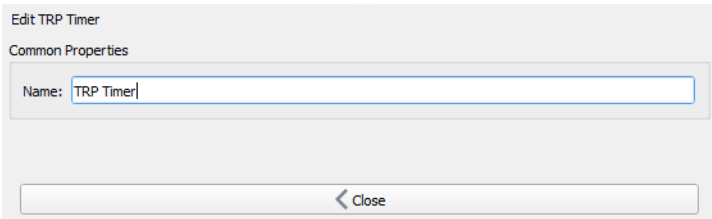


Figure 420 Binding Properties Dialog

| Property | Description |
|---------------------------|---|
| Common Properties: | |
| Name | Enter a unique name for the Binding block. |

Table 80 Binding Properties

Click to jump to [Section C "List of Behaviours and Bindings"](#)

C.2.11 TSL UMD Binding (Group: Multiviewer)

The TSL UMD **Binding** is used to set the properties on a UMD widget from the values in a TSL **Behaviour**.

Two different configuration screens are shown, depending on whether TSL Protocol version 3.1 or 5.0 is selected in the TSL **Behaviour** (property TSL Version).

Edit TSL UMD

Common Properties

Name:

TSL UMD

Source Behaviour:

TSL

Select

TSL v3.1 Options

Left Lamp

Tally 1

Right Lamp

Tally 2

Text Tally

Tally 3

Left On Color

Red

Right On Color

Green

Text On Color

Off

< Close

a) For TSL Protocol 3.1

Edit TSL UMD

Common Properties

Name:

TSL UMD

Source Behaviour:

TSL

Select

< Close

b) For TSL Protocol 5.0

Figure 421 Binding Properties Dialog:
a) For TSL Protocol 3.1.
b) For TSL Protocol 5.0.

| Property | Description |
|--|---|
| Common Properties: | |
| Name | Enter a unique name for the Binding block. |
| Source Behaviour | <p>Select the Behaviour block for the Binding to interact with. This may already be automatically filled out.</p> <p>Click Select for a list of the names of available Behaviour blocks.</p> <p>If required, further select which of the Behaviour's values to use.</p> |
| TSL v3.1 Options: | |
| Left Lamp | Select the value to drive the left lamp tally. (1 to 4) |
| Right Lamp | Select the value to drive the right lamp tally. (1 to 4) |
| Text Tally | Select the value to drive the text tally. (1 to 4) |
| Left On Color | Select the color of choice for use when the selected Tally value is true: Off, Red, Green or Amber. |
| Right On Color | Select the color of choice for use when the selected Tally value is true: Off, Red, Green or Amber. |
| Text On Color | Select the color of choice for use when the selected Tally value is true: Off, Red, Green or Amber. |
| Note: There are no specific TSL 5.0 Binding items. | |

Table 81 Binding Properties

Click to jump to [Section C "List of Behaviours and Bindings"](#)



Grass Valley Technical Support

For details of our Regional Customer Support Offices please visit the Grass Valley web site at:

<https://www.grassvalley.com/contact/support/>

Customers with a support contract should call their personalized number, which can be found in their contract, and be ready to provide their contract number and details.

For technical assistance, contact our international support center, at 1-800-547-8949 (US and Canada) or +1 530 478 4148.

To obtain a local phone number for the support center nearest you, please consult the 'Contact Us' section of Grass Valley's website (www.grassvalley.com).

An on-line form for e-mail contact is also available from the website.

Corporate Head Office

Grass Valley

3499 Douglas-B.-Floreani

St-Laurent, Quebec H4S 2C6

Canada

Telephone:+1 514 333 1772

Fax:+1 514 333 9828

www.grassvalley.com