

Medialon Manager V7

User Guide

Trademark Information

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

Copyright Information

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

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

Document Revision

Date	Document edition	Software version	Revision details	Author/Editor
May 2019	4.4.3 Rev 1	Medialon Manager 6.7 Medialon Showmaster Editor 2.7.0	Published by Barco	Eric Cantrell
June 2019	M515-1	Medialon Manager 6.7 Medialon Showmaster Editor 2.7.0	Re-edited, published by Medialon Ltd	Andie Davidson (ed)
May 2020	M515-2	Medialon Manager 6.7 Medialon Showmaster Editor 2.7.0	Edited with addition of Showmaster Go and Showmaster Mini	Eric Cantrell, Andie Davidson (eds)
June 2020	M515-3	Medialon Manager 6.7 Medialon Showmaster Editor 2.7.0	Terminology changes in software	Eric Cantrell
May 2022	M515-4	Medialon Manager 7.0.0 only	Update to Manager V7	Nate Pease, Andie Davidson (eds)

M515-4

Contents

1 About this Manual	7
1.1 Prerequisites	7
1.2 Objective of this Manual	7
1.3 Conventions in this Manual	8
2 Introduction	9
2.1 Background	9
2.1.1 Programming Environment Features	9
2.1.2 Medialon, an Integrated System	11
2.2 Overview of Medialon Manager	11
2.2.1 Medialon Manager Programming Environment	12
2.2.2 Architecture	14
2.2.3 MxMs	15
2.2.4 Task Engine	17
2.2.5 Graphical User Interfaces	18
2.2.6 Variables	20
2.2.7 Medialon Manager Objects and Interactions	21
2.2.8 Medialon Manager Modes	21
2.2.9 Medialon Manager Product Comparison	22
3 Software Installation	26
3.1 Computer Requirements	26
3.1.1 Minimum System Requirements	26
3.1.2 Suggested System Requirements	26
3.2 Medialon Manager Installation	26
3.2.1 Installation of Main Software	27
3.2.2 Registration	28
4 Programming Environment	30
4.1 Workspace	30
4.1.1 The Workspace	30
4.1.2 Main Toolbar	31
4.1.3 Browser Tab	32
4.1.4 Lister Tab	32
4.1.5 Properties Tab	33
4.1.6 Control Tab	34
4.1.7 Panels Tab	34

4.1.8 Device Map Tab	34
4.1.9 Task Tab	34
4.1.10 Customize Your Workspace	35
4.2 The Finder	37
4.2.1 The Finder Window	37
5 Show Programming	41
5.1 Devices	41
5.1.1 Creating a Device.....	41
5.1.2 Device Properties.....	43
5.1.3 Device Map	44
5.1.4 Testing Devices	45
5.1.5 Device Help.....	47
5.2 Time Based Tasks.....	49
5.2.1 Show Structure	49
5.2.2 Creating a Time Based Task.....	49
5.2.3 Track Management.....	51
5.2.4 Creating Cues.....	52
5.2.5 Selecting Cues	57
5.2.6 Managing Cues.....	58
5.2.7 The Cue Library.....	61
5.2.8 Time X.....	62
5.2.9 Timeline Management	63
5.2.10 Positrack.....	65
5.2.11 Timecode.....	66
5.2.12 Recording Cues	67
5.2.13 Controlling Tasks	70
5.2.14 Views	71
5.2.15 Flash Controls.....	72
5.2.16 Blind Mode.....	72
5.2.17 Lock Task	73
5.2.18 Pause Points.....	73
5.2.19 Tasks in a Project.....	73
5.2.20 Task Setup.....	76
6 Automation Programming	79
6.1 Panels and UserScreens	79
6.1.1 The Panels	79
6.1.2 Creating a UserScreen.....	86

6.1.3 Graphical Objects	91
6.1.4 Graphical Object Properties.....	92
6.1.5 Graphical Object Tools	104
6.2 Variables.....	109
6.2.1 Defining Variables	109
6.2.2 Creating User Variables.....	111
6.2.3 Variable Properties.....	112
6.2.4 Using Variables as Cue Parameters	114
6.3 Expressions	116
6.3.1 Logical Expressions.....	116
6.3.2 The Expression Window.....	118
6.3.3 Operations	121
6.3.4 Function Operators	124
6.3.5 Start Conditions.....	134
6.4 Step Based Tasks.....	135
6.4.1 Creating a step based task.....	135
6.4.2 Selecting Cues	136
6.4.3 Managing Cues.....	137
6.4.4 Flash Controls.....	139
6.4.5 Blind Mode.....	139
6.4.6 Lock Task.....	139
6.4.7 The Cue Library.....	139
6.4.8 Pause Points	140
6.4.9 Tasks in a Project.....	140
6.4.10 Task Setup.....	142
6.5 Device Manager/Showmaster	143
6.5.1 UserScreen Commands.....	143
6.5.2 Task Commands.....	148
6.5.3 Expression	152
6.5.4 Logical Cues.....	152
6.5.5 System Commands	155
6.5.6 Object Commands	158
6.5.7 Manager Variables	163
7 Advanced Programming	166
7.1 Cue Library.....	166
7.1.1 The Cue Library.....	166
7.1.2 Adding Cues to the Cue Library	166
7.1.3 Inserting Cues from the Cue Library.....	167

7.1.4 Testing Cues from Library	169
7.1.5 Deleting Cues from Library	169
7.2 User Groups	170
7.2.1 Creating User Groups	170
7.2.2 Adding Objects to User Groups	172
7.2.3 Sharing Objects.....	174
7.2.4 Smart Groups.....	175
7.3 Networking.....	175
7.3.1 Medialon On Network (MON).....	175
7.3.2 OpenCap.....	176
7.3.3 Medialon On Network Device.....	177
7.3.4 Network Architectures.....	185
7.3.5 Medialon Manager Panel.....	186
7.3.6 Panel/WebPanel/OpenCap Connection Chart.....	187
7.4 Preferences.....	188
7.4.1 General	188
7.4.2 Project	189
7.4.3 Task	191
7.4.4 Run Time.....	193
7.4.5 Network.....	194
7.4.6 UserScreen	195
7.4.7 Watchdog.....	197
7.4.8 Log Traces.....	198
7.5 Advanced Tools	201
7.5.1 Cue Multiple Devices	201
7.5.2 Device Replacement	202
7.5.3 Pause Points	203
7.5.4 Task Execution.....	204
7.6 Import and Export Project.....	205
7.6.1 Importing a Project file.....	205
7.6.2 Exporting / Importing a Task.....	205
7.6.3 Converting a Project	206
7.7 Driver Editing	207
7.7.1 MxM Medialon Low Level Communicator.....	207
7.7.2 Driver Creation	207
7.7.3 User Commands.....	212
7.7.4 Monitoring, Request and Variables	215
7.7.5 Save and Share Drivers	220

7.8 User Mode.....	221
7.8.1 User Mode Rights.....	222
7.8.2 Task Editor in User Mode.....	222
8 Menus and Wizards	224
8.1 Menus	224
8.1.1 File Menu.....	224
8.1.2 Edit Menu	225
8.1.3 Object Menu	227
8.1.4 Windows Menu	230
8.2 Drag and Drop Wizards	234
8.2.1 Dragging onto Graphical Objects	234
8.2.2 Drag and Drop on Tasks	237
8.3 Task Creation Wizards	239
8.3.1 Graphical Object Wizards	239
8.3.2 Variable Wizards.....	241
8.4 Keyboard Shortcuts	242
8.4.1 File main menu	242
8.4.2 Edit main menu.....	242
8.4.3 Object main menu	242
8.4.4 Windows main menu.....	243
8.4.5 Help main menu	243
8.4.6 Workspace navigation.....	243
8.4.7 Browser Tab Navigation.....	243
8.4.8 Lister Tab Navigation.....	244
8.4.9 Properties Tab Navigation	244
8.4.10 Task Editor Tab Navigation.....	244
8.4.11 Time X window navigation	246
8.4.12 UserScreens manipulation	246
8.5 Action on Double-click.....	246
8.5.1 Cue double-click	246
8.5.2 Variable double-click.....	246
8.5.3 Device Map double-click	247
8.5.4 Task double-click.....	247
8.5.5 Ctrl + Task double-click.....	247
8.5.6 Device double-click	247
8.5.7 Ctrl + Device double-click.....	247
8.5.8 Command double-click	247
8.5.9 Panel double-click	247

8.5.10 UserScreen double-click	247
8.5.11 User Group double-click.....	247
9 Manager V7 Specific Features.....	248
9.1 Overview of Medialon Manager	248
9.1.1 Medialon Manager V7	248
9.1.2 Medialon Manager Modes.....	249
9.1.3 Medialon Manager V7 License Options	249
9.2 Resources	250
9.2.1 Resource Management (MRC).....	250
9.2.2 MRC Help	261
9.3 Medialon Watchdog.....	262
9.3.1 Watchdog Overview	262
9.3.2 Medialon Watchdog Window	262
9.4 Project Files	266
9.4.1 Project Files and Folders.....	266
9.4.2 Recovering the Previous Version of a Project.....	266
9.4.3 Opening Project from previous version of Medialon Manager	267
9.4.4 Upgrading Manager V7, MRCs and MxMs	267
Annex A: Medialon Resource Connector	269
Embedded Resources	269
External Resources	269
Annex B: Medialon MxMs	276
Service MxMs.....	276
Device MxMs.....	278
Annex C: Regional and Language Options	291
Configure Windows to type in multiple languages	291
Contact Us	292

1 About this Manual

This manual is written for professional users: show technicians, freelancers and end user technicians who are responsible for programming and running show control systems in live events or permanent installations.

It provides readers with the technical information required to set up and program a Show Control installation based on Medialon Manager.

Important: please read:

- M562 Medialon Safety Instructions for installation, use and service
- M561 End User's License Agreement for Medialon Software and Plug-Ins. Warranty Terms and Conditions for Medialon Hardware

For Showmaster show controller hardware installation and maintenance, please refer to:

- M569 Medialon Showmaster Hardware Manual

1.1 Prerequisites

This manual assumes the following prerequisites:

- Proficiency using Windows interface
- General knowledge of computer hardware
- General knowledge of control protocols: DMX 512, MIDI, Serial, Timecode
- General knowledge in audiovisual: Video, Video-processing, Audio, Lighting, Machinery
- General knowledge in show structure

1.2 Objective of this Manual

After reading this manual, you will be able to:

- Set up software and hardware required for a Show Control installation using Medialon Manager
- Program shows
- Develop custom user interfaces and logical treatments for complex show control applications
- Run and maintain a Show Control installation based on Medialon Manager

1.3 Conventions in this Manual

This manual uses the following conventions:

[<link>](#) Each time you find text written in blue, it corresponds to an active link to an internal reference or to a website.



This icon indicates notes or special attention.



This icon in the left margin means the text describes a feature only available in LocalPanel.



This icon in the left margin means the text describes a feature only available in WebPanel.

2 Introduction

This section describes Medialon Manager, what its main elements are, and how they interact with each other.

This section also helps those who are new to Medialon Manager to understand its core engine and ease their understanding of the following chapters.

2.1 Background

The introduction of our first Audiovisual Control Software in 1999 was driven by our vision to develop innovative technologies to help creative people to achieve compelling entertainment and edutainment experiences.

From the original Medialon Manager Control software, a range of products have evolved: Showmaster Pro, Showmaster LE, Showmaster Mini, and Showmaster Go controllers all using Medialon Manager V7 and Medialon Manager V7 Lite programming environment.

Showmaster controllers have the advantage of being all-in-one devices that include hardware interfaces such as DMX, MIDI, Serial, Timecode, Digital IOs and of course TCP/IP.

Medialon Manager provides easy integration in an IT environment when a link to databases or other IT resources is needed. It can also be deployed on a virtual machine, making it hardware independent, and can be sized to your exact needs by adding Medialon or third-party hardware interfaces.

Thanks to the easy graphical programming environment, show control applications are programmed in much less time than with any other conventional control system; neither coding nor compilation are required.

The native synchronization capabilities and timeline with drag and drop programming makes Medialon Manager the ideal choice for show control applications. However, complex logical task programming is also available, allowing your shows to be interactive and respond to input from operators as well as guests.

2.1.1 Programming Environment Features

A driver for virtually anything ...

Medialon Manager uses plug-in drivers called MxMs (Medialon X Object Modules) to control devices. A wide range of MxMs are included for most popular devices on the market connected through Serial, MIDI, Timecode, Infrared, Modbus, TCP/IP, etc. The programming environment also includes a powerful tool to write your own drivers for serial, TCP/IP, UDP and MIDI devices. These custom drivers can be shared on Medialon's website with other Medialon users.

Not just a device controller ...

Medialon Manager includes MxM plug-ins to add functionalities and to control software devices found in modern AV/IT installations, such as SQL Databases for data management, FTP Client for media management and transfer, email engine, and SNMP for monitoring and controlling network switches and IP cameras.

Medialon Manager really can take advantage of the power of modern integrated AV/IT (depending on the hardware used).

Device synchronization

Timelines with drag and drop programming allow synchronization of virtually any device to external or internal timecode with an accuracy of a 0.01 sec.

Timelines use a feature unique to Medialon, Positrack, which locates any device at the position it should be at any given time when moving back and forth in the show. This makes rehearsal much easier and efficient.

Timelines offer the ability to record external signals such as DMX, MIDI, and serial for later synchronized playback.

Logical programming

Control programming always requires more than a timeline.

Medialon Manager offers both the time-driven programming and event or external condition-based programming capabilities.

Programming can be triggered by input from a touch screen, digital I/Os, or data coming from controlled devices. Variables can be used throughout to perform complex logical programming.

An open network application

Medialon Manager has been network oriented from its original version. Not only does it control many AV and IT devices over TCP/IP, but it also communicates with other Medialon systems on the same network (or even across the Internet).

The Medialon On Network (MON) protocol publishes tasks and commands to other Medialon Controllers for easy development of shared applications. MON is also available as a C++, C#, or raw XML via TCP API, offering a powerful interoperability tool to communicate with third party software or develop add-on modules or interfaces.

Panels and graphical UserScreen interfaces

User interfaces can be displayed on local or remote panels within web browsers. Medialon Manager V7 software also includes Local Panels, or UserScreens built into the control software and displayed locally on the machine running Medialon Manager V7.

UserScreens can include graphical objects such as buttons, sliders, LEDs, bar graphs, text displays, lists, pictures and more for building a fully customized control system interface. UserScreens can also be served as web pages accessible from web browsers on all kinds of devices connected to the same network.

Hardware independent

Independently from the plug-in drivers, Medialon Manager uses 'resources' to physically connect to the controlled devices. Resources can be local resources from PCI Cards installed in a computer running the Medialon Manager software, or an IP-based solution such as serial over IP converters, or other control hardware such as Artistic Licence Art-Net, Global Caché, etc. Networked extensions can be used for both the software version and embedded controllers.

2.1.2 Medialon, an Integrated System

Every Medialon controller can share tasks and variables on the network, allowing other controllers access and control them. It is possible to have the entire program spread over several controllers. Different topologies are possible, depending on the complexity or required functionalities.

2.2 Overview of Medialon Manager

Medialon Manager and Medialon Manager Lite provide the programming environment for systems of controllers including:



Showmaster Go



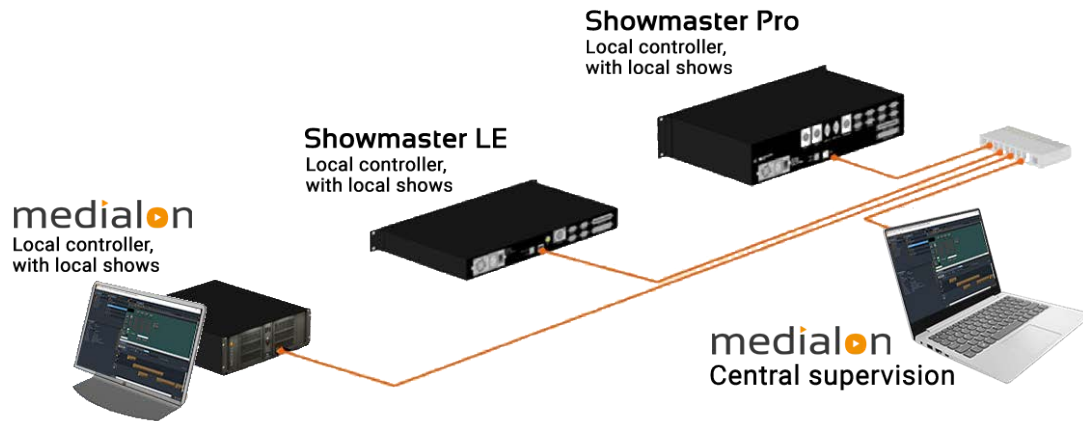
Showmaster Mini



Showmaster LE



Showmaster Pro



2.2.1 Medialon Manager Programming Environment

Medialon Manager controls and synchronizes digital audio and video, lighting dimmers and desks, matrix switchers, image processors, video walls, lasers, fireworks, animatronic figures, doors, and special effects, all over one network (or many networks, depending on requirements). It offers frame accurate synchronization, DMX recording and playback, multitasking, and device position tracking. These tools open a lot of creative control options.

Device control

Visually organized by the Device Map, Devices are the metaphor for controlling any type of equipment via serial, TCP, UDP, MIDI, DMX, IR, Relays, or Analog I/O. Medialon Manager uses plug-ins called MxMs (Medialon X Object Modules) to control devices. There are a wide range of MxMs for most popular devices on the market communicating via serial, MIDI, Timecode, Infrared, Modbus, and TCP/IP. Medialon Manager also includes a powerful tool to write your own drivers using serial, TCP, UDP, and MIDI. Custom drivers can be reused and shared with other Medialon users.

Each MxM provides a set of commands that can be sent to the devices and returns a set of variables containing device feedback which can be used in programming to monitor the device's status or to make automated decisions. Some MxMs come with a built-in graphical user interface called a Device Control Panel, which can be used to control and monitor devices without requiring any programming to build the user interface.

Medialon Manager has been designed to natively synchronize the devices and controls with an accuracy of one hundredth of a second (10 milliseconds).

Medialon has always been hardware independent; it can use virtually any hardware to control devices. Independently from the plug-in (MxM), Medialon Manager uses 'resources' to physically connect to each device. Resources can be serial ports, PCI Timecode card in a computer, a relay on a Showmaster V2, an IP serial converter, or any other hardware such as AMX® Netlinx interfaces, Artistic Licence Art-Net, Global Caché, etc. The Medialon Resource Connector (MRC) manages and configures the necessary resources for Medialon Manager projects.

Time based (timeline) tasks

Timelines are the core show engine in Medialon Manager. The timeline is a time metaphor in which users program cues to control various devices. The timeline offers tools to ease programming and playback of the show.

Positrack is the capability for the software to reposition every device at the position or state they should be at when jumping back and forth in the timeline. Positrack allows fine-tuning of a complex show without needing to run the show from the beginning each time you want to test a new adjustment.

The Task Control Panel is a simple user interface to control each timeline. It can be enlarged and sent to another monitor (for the lighting designer, for example). The Task Control Panel window provides start, pause, and stop buttons, the current timecode of the timeline, a display for comment and countdown cues. Comment cues are user defined cues in the timeline to add comments to programming, Countdown cues give a comment with a countdown to the next action in the show.

Timelines can also be viewed as a list, which provides access to 'Flash Controls'. This functionality allows users to execute cues out of order in the show as needed. While the show is running, the user can fire cues in advance or disable upcoming cues. This gives extreme flexibility in live event shows where things on stage sometimes do not happen as expected.

Several timelines can run simultaneously, a timeline can control another one, or be synchronized to another one or run independently, and all the timelines can be synced to external timecode.

Step based tasks and variables

Step based tasks perform cues, just like the timelines, but in a 'step by step' manner without reference to time. Step based tasks can also perform logical operator cues such as 'If, Then, Else', 'While, End While' and 'Wait For'.

Variables returned by devices, as well as user-defined variables, can be used in step based tasks. Variable listers show the values of all variables in the project.

Step based tasks can be used to create powerful programming that performs complex actions or makes decisions.

Panels and user interfaces

Although the software comes with a number of built-in control panels to control devices or timelines, an additional UserScreen builder allows programmers to define their own screens with buttons, sliders, LEDs, bar graphs, text displays, lists, pictures and more. This feature allows the creation of a 'protected' user interface, hiding the programming interface from show operators. These user interfaces can be opened inside local screens or be displayed in web browsers on devices elsewhere in the network.

Supervisor / Agent capabilities

Medialon Manager is network-oriented by design. Medialon Manager applications can share their objects (like variables, tasks, and UserScreens) on the TCP/IP network (Agents), or see variables and send commands to other instances of Manager (Supervisor). It is easy to build networked applications where control needs to be distributed to several other devices running Medialon Manager, such as touch panels. Another common topology includes a central system controlling and supervising other systems or touch panels.

Add-on functionalities

Some advanced functionalities are not needed in most applications but are available by using special MxMs. The Files Management MxM has all the necessary tools to copy, move, rename, and delete files in a Windows environment. The Strings Management MxM provides conversion and analysis of text strings. The Event Log MxM allows the programmer to define what events should be logged in the system and generates text files that can then be emailed or stored to network attached storage. Most of these functionalities are useful in permanent installations more than for live events.

Internet functionalities





Medialon Manager V7 includes add-on functionalities in the form of specific MxM plug-ins to perform actions which are related to internet communication. Most of these functionalities are useful in permanent installations more than for live events. The Email MxM allows email to be sent or received. The FTP Client MxM lets Medialon Manager handle all kind of FTP transfers. The Database MxM provides an interface with SQL type databases configured as data sources in Windows' ODBC settings, including simple data sources like text or CSV files, or complex database engines like MySQL, PostgreSQL, SQLite, Oracle, or Microsoft SQL Server.

Sophisticated Programming

Cues can perform actions that act upon the Medialon Manager project itself, like 'Pause' or 'Synchronize Task', to make real-time changes during the show. Significant internal or device conditions can be identified as variables, which can be used locally or published on the network for other Medialon Manager systems or for external display or control. Variables can be continuously monitored and combined to fire tasks. It is possible to start tasks based on device exceptions, day, date, time, or outside authorization.

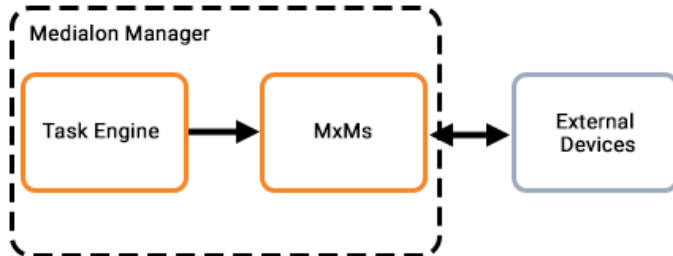
2.2.2 Architecture

Medialon Manager V7 uses 4 main objects for building and creating a project:

-  Devices (MxMs)
-  Tasks
-  Panels
-  Variables

2.2.3 MxMs

Medialon Manager sends commands to external devices via software modules called Medialon X Object Modules (MxMs). Medialon develops these software plug-ins to allow Medialon Manager to control external devices or increase its functionalities. In return, the MxMs receive data from the external devices and make it available for Medialon Manager.



All MxMs are available from 7thSense (see <https://portal.7thsense.one/medialon-help/>). There are two major types of MxMs:

- Device MxMs.
- Service MxMs.

Device MxMs

MxMs that communicate with external devices are referred to as 'Device MxMs'. Device MxMs allow Medialon Manager to communicate with hardware machines such as video projectors and lighting consoles, but they can also control software such as Dataton Watchout. A wide range of Device MxMs are available for your show control application.

- Video Servers (Serial, TCP/IP)
Doremi, Turbo, Mediasonics, Dataton Watchout, Windows Media Player, Videro media player software
- Video Processors (Serial, TCP/IP)
Barco Encore, Analog Way iX Series, etc.
- Audio Consoles and Audio Processors (MIDI, TCP/IP)
BSS Soundweb London, QSC Q-Sys, Yamaha O1V, etc.
- Lighting Consoles (MIDI, TCP/IP)
GrandMA, ETC EOS, Martin MaxXYZ, Wholehog, etc.
- Dimmers, Lighting Fixtures (DMX)
Dimmers, DL3s, MAC 2Ks, Foggers, etc.
- Media Servers (DMX, TCP/IP)
Catalyst, Coolux, 7thSense Delta, Disguise D3, etc.
- Switchers and Routers (Serial, TCP/IP)
Barco, Autopatch, Extron, etc.

- Video Projectors (Serial, TCP/IP)
Barco, Christie, Sony, Sanyo, etc.
- Digital I/O and PLCs (Serial, TCP/IP and dedicated PCI boards)
Advantech ADAM, Opto-22, Beckhoff, onboard digital I/O ports, Modbus, OPC, etc.


Service MxMs

Some MxMs implement services as extensions of Medialon Manager and provide sophisticated functionalities like scheduling, error logging or database management. They are referred to as 'Service MxMs'. These MxMs greatly enhance Medialon Manager and provide the necessary resources to manage and control information or perform special tasks.

Data management

- *Database*: This MxM provides an interface with SQL compatible databases via the Windows ODBC mechanism. It allows reading, modification and creation of data and records, and sending of SQL requests to the database.
- *Files Management*: This MxM provides a simple way to copy, move or erase files on a hard disk or a Windows network. It also provides commands for reading and writing files.
- *String Management*: This MxM provides conversion and analysis of character strings.

Utilities

- *Email*: This MxM handles the sending and the reception of emails in a Medialon Manager project. It has all the functionalities of regular email software, including management of modem connections.
-  *Event log*: MxM EventLog is designed to produce messages and event reports. It provides functions to store, sort, export and print event lists by date and time.
- *Network Utilities*: Network Utilities MxM provides several commands for network management and system information retrieval.
- *Variables Management*: This MxM provides commands that act on variables, and these commands are Positracked when they are used in a Medialon Manager timeline.

Low Level Communicator

The Low Level Communicator MxM is a low level and powerful MxM. It is a low-level MxM because it is not dedicated to a specific device and is able to use many raw communication protocols.

You can write your own set of commands, create monitored variables linked to the reception of specified frames, and much more to create drivers dedicated to your specific devices and projects.

Five types of communication are available:

Introduction

- Serial
- TCP Client
- TCP Server
- UDP
- MIDI

Protocols and hardware resources

MxMs have been developed for all standard communication protocols:

- Serial RS232, RS422 protocols
- TCP/IP and UDP/IP communication
- DMX512 via a PCI Card, Showmaster hardware, Art-Net, sACN
- MIDI using PC Audio Cards, USB Interfaces, or Showmaster hardware
- Timecode via a PCI Card, USB Interfaces, or on a Showmaster Pro
- Infrared via GlobalCache or AMX Netlinx
- Modbus/TCP
- OPC


Independently from the MxM, Medialon Manager uses a 'Resource' to physically connect to the device. Resources include Windows COM ports, Showmaster V2 relay outputs, Serial over IP converters, or any other hardware such as AMX® NetLinx interfaces, Artistic Licence Art-Net, Windows® PC resources, Global Caché, etc. Medialon Resource Connectors (MRC) are used to configure the necessary resources for your projects.

2.2.4 Task Engine


The core of Medialon Manager is its task engine which allows the execution of programmed cues.

How and when cues are executed depends on whether the cues are in a time based task or a step based task. Medialon Manager can run these tasks in parallel and concurrently, thanks to its built-in multitasking architecture.

Time based (Timeline) tasks

In time based tasks , cues are executed based on the current time position of the task. This uses a familiar Show Control concept of the 'timeline', where cues are laid down on tracks and are triggered as soon as the time of the task reaches the cue.

Step based tasks

In step based tasks , there is no notion of or reference to time. Cues are executed one after the other in a 'Step by Step' manner. The task flow can be modified by conditional statements, much like a traditional scripting or programming language, making them useful to implement logical behaviors.

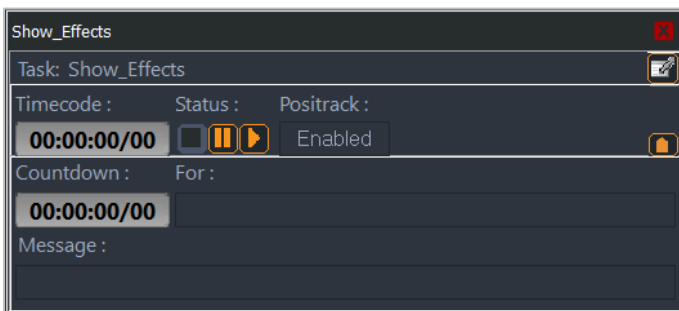
2.2.5 Graphical User Interfaces


Medialon Manager provides two types of user interface windows which can give you immediate control and feedback of what is going on while your show is running:

- Built-in User Interfaces provided by Medialon Manager or by the MxMs.
- Custom User Interfaces which you can create and configure depending on your needs.

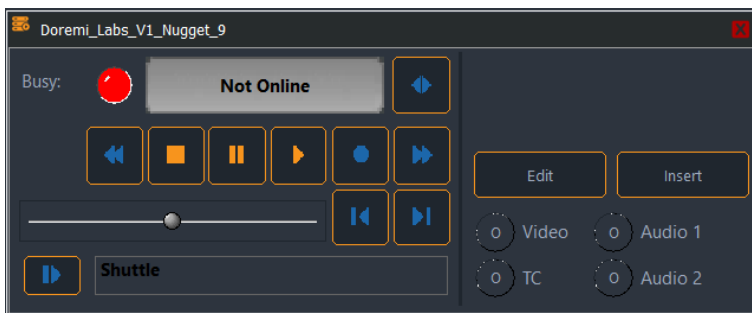
Built-in User Interfaces

Task Control Panels



Exclusively designed to control time based tasks, the Task Control Panel  displays information about a task including the current timecode position of the task as well as countdowns and messages attached to the next cue to be executed.

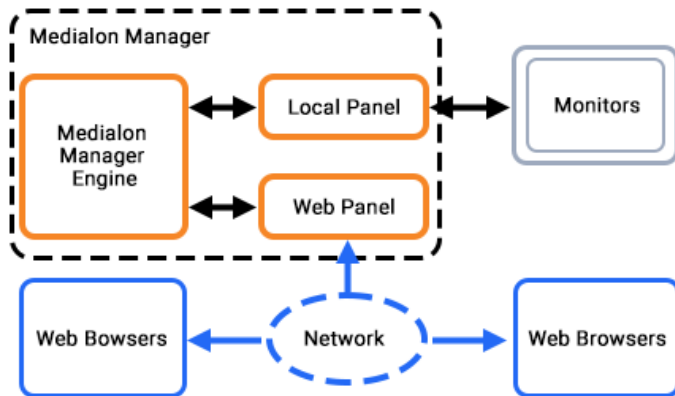
Device Control Panels



Device Control Panels look like equipment front panels and feature the necessary buttons and displays needed to control your devices. Device Control Panels may be opened either in the Control Center window, or in independent floating windows.

Custom User Interfaces

Medialon Manager V7 provides 2 kinds of panels  to host your user interface:



- *Local Panel*: Your user interface is displayed locally on monitors connected to Manager V7 host PC.
- *Web Panel*: User interface, opened with a web browser on any remote machine connected to Manager V7.

Design of these panels is done directly in your Medialon Manager V7 project.

Custom user interfaces called UserScreens can be easily built thanks to the graphical toolset provided by Medialon Manager. Graphical objects on UserScreens can trigger tasks and/or display system and device information, thus allowing you to design complete custom user interfaces.




Before you can type in a language other than the one installed in Windows, you must configure the keyboard to match the language. See also [Regional and Language Options](#).

2.2.6 Variables


The various elements of Medialon Manager, such as tasks, devices and UserScreens share information through variables. Variables can be described as objects that contain values.

There are 6 types of variables used by Medialon Manager and the MxMs to store information. Each type has its own specificities and parameters.

 Integer

 Real


 String

 Enum

 Time

 Date


System Variables

Variables that are automatically created by Medialon Manager are called System variables . For example, there are variables for the current time, Manager's current status or project file location.

System variables are automatically created with the devices. These may be the current timecode of the device, whether the device is online or the list of clips available on a video server.


Tasks may also have variables, such as Status or TimecodeStatus for time based tasks. These System variables are not automatically created; you must define their creation upon task creation, as part of your Medialon Manager preferences or by editing the task's settings.

Screen Variables

These are variables that are automatically created with Panels and UserScreens , such as position of the mouse pointer or last key pressed.

All dynamic objects present in a UserScreen generate one or more variables. Pushing a button changes its Click variable's value and its Status variable. Changing variables can trigger a task to start. A text display can be updated by a changing timecode variable, for instance. Screen variables are 'Read/Write' and can be updated manually or through programming.

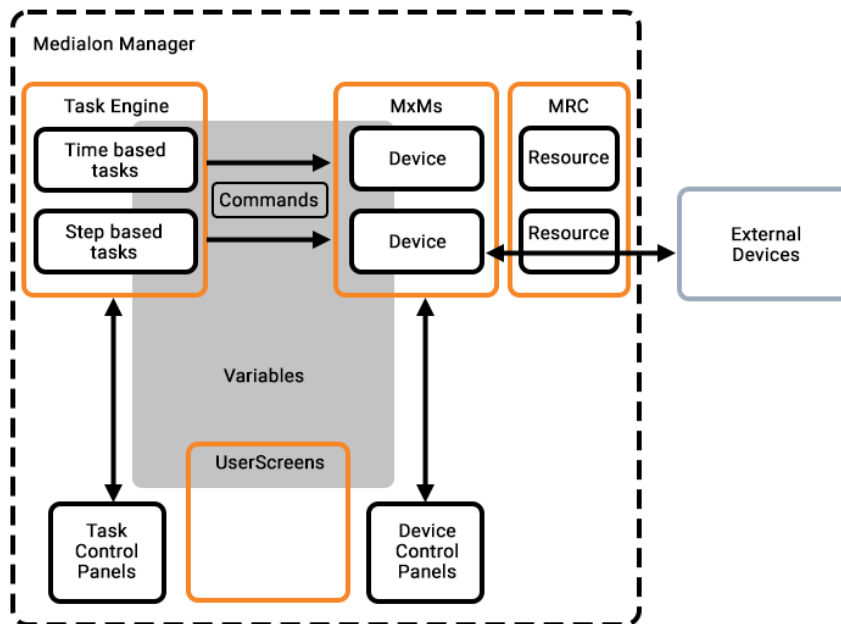
User Defined Variables

You may also create your own User variables  to store information. These can be of any type and all the parameters are editable. User variables are 'Read/ Write', meaning they can be updated manually or by programming. They may also be made persistent. Variables with the

Persistent property enabled have their values stored to a file so the value is retained through a reboot of the show controller.


2.2.7 Medialon Manager Objects and Interactions

The following diagram outlines the interactions between the objects inside Medialon Manager. Notice that a key element of Medialon Manager is the variables, which allows all the different elements to share their information in the system.




2.2.8 Medialon Manager Modes

Stop Mode

The first time you launch Medialon Manager, you will be in Stop Mode . In this mode, Medialon Manager does not communicate with any external equipment, nor does it perform any internal tasks. Stop Mode is the default programming mode where your project is built. Devices, Tasks, UserScreens and variables can be created, modified, and deleted freely.


Pressing the *Esc* key or clicking on the Stop Project button will return you to Stop mode if the project is running Debug, User, or Run mode.

Debug Mode

In Debug Mode , Medialon Manager fully interacts with external devices. The programming interface, including the Browser, Lister, Properties, and Task Editor panes are still accessible. This allows you to see what is happening internally while your project runs, allowing you to make adjustments without stopping the project.


Debug Mode is accessible from Stop Mode by typing the *F2* key or clicking on the Debug Project button.

User Mode

Medialon Manager's User Mode gives the opportunity to the programmer to decide which objects are accessible to the user (tasks, devices, command for devices, variables). The User Mode  works like the Debug Mode, but only with the preselected objects.

User Mode is accessible from Stop Mode by typing the *Ctrl+F2* key or clicking on the User Mode Project button.

Run Mode

Run Mode  is the final runtime mode of a project. Medialon Manager fully interacts with external devices and hides the programming interface. Only UserScreens created in the LocalPanel and Device Control Panels and Task Control Panels are displayed.

Run Mode is accessible from Stop Mode by typing the *F3* key or clicking on the Run Project button.

The three running modes (Debug, User, and Run) can only be entered from Stop mode. It is not possible to switch between Debug and Run mode directly.

2.2.9 Medialon Manager Product Comparison

Medialon Manager has evolved into a range of products, from embedded controllers to open platform control software. The chart below describes the main features available for each Medialon Manager product.

	Embedded Controllers				Control Software (PC or Virtual Machine required)	
	Showmaster Mini	Showmaster LE	Showmaster Pro	Showmaster Go	Manager V7 Lite	Manager V7 Pro
Hardware Features						
Serial	2	2	8	w/ ext. ⁱⁱ		
USB	4 × USB 3.0	2 × USB 3.0	2 × USB 3.0	3 × USB 2.0 (Type-A) 1 × USB 3.1 (Type-C)		
Video Output	2 × HDMI UHD	–	–	4 × DisplayPort™ 1.2		
Digital Inputs	–	4	16	3 ⁱ		
Relay Outputs	–	4	16	3 ⁱ		
MIDI	–	1	1	–		
DMX	w/ ext. ⁱⁱ	1 physical 5-pin	2 physical 5-pin	w/ ext. ⁱⁱ		
DMX Extension max	1 Art-Net/sACN	4 Art-Net/sACN	20 Art-Net/sACN	Unlimited	Unlimited	Unlimited
Ethernet	2	2	2	2		
Timecode	–	–	1 in & 1 out	–		
Infrared Ports	w/ ext. ⁱⁱ	w/ ext. ⁱⁱ	w/ ext. ⁱⁱ	w/ ext. ⁱⁱ	w/ ext. ⁱⁱ	w/ ext. ⁱⁱ
Third Party Hardware Ext.	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited
Software Features						
Programming Environment	Core M6	Core M6	Core M6	Core M6	Core M6	Core M6
Timeline Programming	–	1	Unlimited	1 (Lite) Unlimited (Pro)	1	Unlimited
1/100th Second Sync Accuracy		✓	✓	✓	✓	✓
Device Positracking		✓	✓	✓	✓	✓
Device Recording (DMX/MIDI/Serial/TCP/UDP)	✓ ⁱⁱⁱ	✓	✓	✓	✓	✓
Logical Programming	✓	✓	✓	✓	✓	✓
Real Time Program Testing	✓	✓	✓	✓	✓	✓
Drivers Editor	✓	✓	✓	✓	✓	✓
Local Media Playback	✓	–	–	✓	✓	✓
Dante Audio Output	–	–	–	16 / 32 / 64 channels	–	–
WebPanel / Manager Panel / OpenCap Connections	1	3	10	2 (Lite) 10 (Pro) ^v	2 ^{iv}	10 ^{iv}
All other Medialon on Network Connections ^v	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited
Local Panel	✓	–	–	✓	✓	✓
Log Files	✓	✓	✓	✓	✓	✓
SQL Database	–	–	–	✓ ^{vi}		✓

ⁱ Showmaster Go has 3 channels of General Purpose I/O, meaning they can be software configured as input or output.

ⁱⁱ w/ ext. – with third party hardware extension. Please refer to Technical Specs for exact compatibility per product.

ⁱⁱⁱ Device Recording is limited to the resources available. On Showmaster Mini, MIDI resources are not available, so MIDI recording is not available.

^{iv} For Medialon Manager software, WebPanel / Manager Panel / OpenCap Connections can be increased in batches of 10.

^v These connections may be limited if the client application has its own limitations, such as Manager Panel.

^{vi} With Manager Pro only.

Medialon Showmaster Mini

Showmaster Mini is a capable and flexible device is suited to simple AV control applications and installation types. Its ruggedized, fanless, small form factor makes it easy to mount where AV racks are not suitable. Showmaster Mini carries a reduced-feature version of Medialon Manager show control software, making it a low-cost solution that still carries the power, flexibility, and ease of use Medialon is known for. Designed for less demanding applications, it starts with a 20-device limitation, expandable in 10-device batches up to a max of 50 devices. This show controller runs Medialon Manager V7 for Showmaster Mini for step-based tasks only.

Medialon Showmaster LE

Showmaster LE is an all-in-one Embedded Show Controller fitted with all necessary show protocol interfaces. It controls and synchronizes dimmers, lighting desks, video projectors, video servers, sound processors etc. Showmaster LE also provides digital I/O.

Medialon Showmaster Pro

Showmaster Pro is an all-in-one embedded Show Controller fitted with all necessary show protocol interfaces. It controls and synchronizes dimmers, lighting desks, video projectors, video servers, sound processors, and many other types of equipment. Showmaster Pro also provides onboard digital I/O and balanced Timecode input.

Showmaster Go

Showmaster Go is a ruggedized, small form-factor device, ideal for ride vehicles and parade floats, Showmaster Go combines a Pro version Medialon show controller with 16 channels of Dante network audio output (expandable). Running Medialon Manager show control software, this little box can control audio, video, lighting, animatronics, special effects, interfacing with PLCs and ride controllers, and synchronizing to multiple time sources.

Medialon Manager V7 Lite

Medialon Manager V7 Lite is designed for small to medium size permanent installations. Though limited in capacity, it offers all Medialon Manager control and programming power at an affordable price. It only supports one timeline, but allows 3rd party extension interfaces, two simultaneous panel connections and signal recording for Serial, DMX, MIDI, and TCP/IP.

Medialon Manager V7 Pro

Medialon Manager V7 Pro is designed for any show and media control application, from a simple stand-alone to large networked systems. Several Manager V7 stations can be connected together and share data over an Ethernet network. It is perfect for demanding applications such as theme parks, museums, or complex live events. Medialon Manager V7 Pro offers multiple time based tasks (or timelines), step based tasks and custom control panels (UserScreens). It also allows ten simultaneous touch panel connections and signal recording for Serial, DMX, MIDI, and TCP/IP.

Medialon Manager V7 Panel

Medialon Manager V7 Panel is specifically designed to implement remote control panels, typically on a touch screen computer. This free version of Medialon Manager V7 cannot control external devices but is able to connect and share data with other Medialon Manager systems on the network.

3 Software Installation

In this section you will discover how to install Medialon Manager on your computer.

3.1 Computer Requirements

3.1.1 Minimum System Requirements

- 1.7 GHz PC Core i3 Processor
- 2 GB Memory
- 40 GB Hard Disk
- Windows 7 or higher (Windows 10 preferred)*

* Medialon Software Products are compatible with Windows 7, 8, 8.1, 10 in 32-bit and can be used on these platforms for all kind of use. They can run on Windows 64-bit platforms (Windows 7, 8, 8.1 and 10) in 32-bit compatibility mode.

3.1.2 Suggested System Requirements

- 2.5 Ghz PC Core i7 Processor
- 4 GB Memory
- 60 GB Hard Disk
- Windows 10 Enterprise LTSC
- Dedicated HD-1080 Graphics Card
- Sound Adapter

3.2 Medialon Manager Installation

This chapter describes the procedure to follow to install Medialon Manager V7 on your computer. Please follow it closely, as correct installation of Medialon Manager is very important for the success of your project.

The Medialon Manager Installer includes:

- Medialon Manager V7
- Medialon Manager V7 Panel
- Medialon mLearn Infrared driver creation software
- Medialon Product Browser software

- Medialon Manager Reference Manual
- Runtimes for MxMs

3.2.1 Installation of Main Software

Medialon Software Products are compatible with Windows 7, 8, 8.1, 10 in 32-bit and can be used on these platforms for all kind of use. They can run on Windows 64-bit platforms (Windows 7, 8, 8.1 and in 32-bit compatibility mode).

To install Medialon Manager V7 software:

1. First make sure you have administration rights on your computer.
2. Set the license dongle aside; do not connect it to the computer yet.
3. Download the installer to your computer and double-click mSetup.



4. Click 'Install Medialon Manager'.
5. Click 'Next' on the Welcome page.
6. Read the Software license Agreement, mark 'I accept the agreement', then click the 'Next' button.
7. Click 'Next' on the Choose Destination Location page. This will install Medialon Manager V7 in the default Medialon Manager directory.
8. Check the box for 'Set up the MedialonManager.exe to be launched as administrator' and click Next
9. Verify settings and click Install
10. The installation process copies the Medialon Manager V7 files.
11. Click 'Finish' to complete Medialon Manager installation. Then click Quit to close the installation screen.
12. The license dongle can now be connected and Medialon Manager can be launched.

Software Installation

If you have a beige, grey, or black dongle, use the following procedure to install the driver for the dongle:

1. Open Windows Device Manager from Control Panel/System/
2. Select USB Device node
3. Select SmartKey device
4. Go for Update Driver
5. Select option 'Choose file location'
6. Driver files are located in 'C:\Windows\SysWOW64' folder.

If you do not have a driver please contact support@7thsense.one.

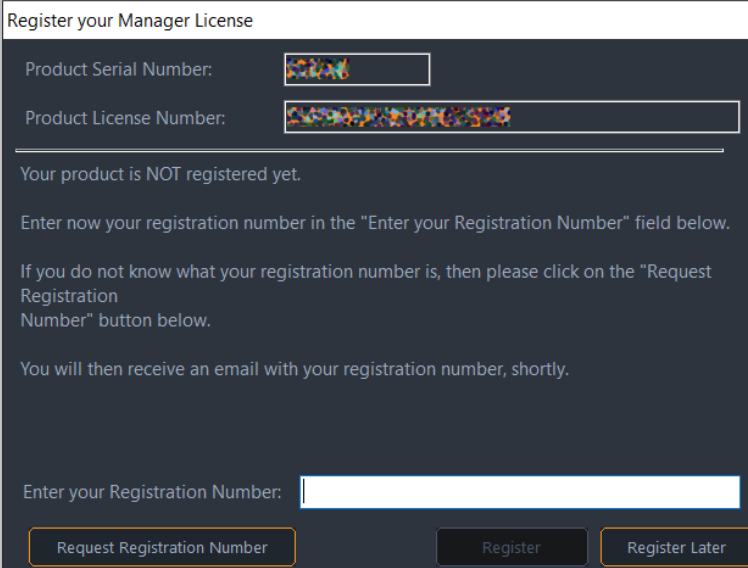
3.2.2 Registration

Medialon Manager will ask for your registration number upon startup. Please take a few moments to register your product with us.

If you do not yet have your registration number, please use one of the following methods to obtain it:

- Click on 'Request Registration Number' button if you have an internet connection. An email will be sent to you with your registration number.
- Complete the online software registration form at www.medialon.com/registration.

Once you have received your registration number, enter it into the registration number field and click the 'Register' button.



The screenshot shows a registration dialog box titled "Register your Manager License". It contains two input fields: "Product Serial Number" and "Product License Number", both filled with random characters. Below these fields, a message states: "Your product is NOT registered yet. Enter now your registration number in the 'Enter your Registration Number' field below. If you do not know what your registration number is, then please click on the 'Request Registration Number' button below. You will then receive an email with your registration number, shortly." At the bottom, there is an input field labeled "Enter your Registration Number:" and three buttons: "Request Registration Number", "Register", and "Register Later".

You do not need to register Medialon Manager V7 to use it. If you are waiting to receive your registration number from Medialon, you can click the 'Register Later' button to close the

Software Installation

Registration dialog box and continue using the software. The dialog box will reappear each time Medialon Manager V7 is launched.

Should you need to use your license on several computers, you may use this number to register your software on all of them. You will, however, need the dongle key to run Medialon Manager.



We strongly recommend you register your license if you start your project using the 'autorun' option. The registration dialog box will continue to be displayed each time you launch your project until the software is registered.

3.3 Medialon Product Browser

This is installed with Medialon Manager and with Showmaster Editor. Run it to show all Medialon devices connected:

The screenshot shows a window titled "Medialon Product Browser 7.1.0" with a menu bar containing "File" and "Help". Below the menu bar is a table with the following columns: "Device Name", "Type/Model", "Version", "Address", and "Description". The table lists several devices, including App Remote devices, Showmaster Pro and XS devices, and Medialon Audio Servers. A "Connect" button is visible at the bottom left of the window.

Device Name	Type/Model	Version	Address	Description
Delta4298	App Remote	6.1.3	10.0.0.194:4550	Medialon AppRemote
FLPGBLN	App Remote	6.1.3	10.0.0.66:4550	Medialon AppRemote
ML-LT-001-V6	App Remote	6.1.3	10.0.0.253:4550	Medialon AppRemote
SMPPro_2303	Showmaster Pro	2.7.3-CAL_369C55f	10.0.0.52:4256	Showmaster Core
SMLE_2207	App Remote	6.1.3	10.0.0.143:4550	Medialon AppRemote
XS-20100002	Showmaster XS	2.7.3	10.0.0.94:9255	Manager Runtime
col-10yo	App Remote	6.1.3	10.0.0.96:4550	Medialon AppRemote
XS-20100002	Showmaster XS	2.7.3	10.0.0.94:4256	Showmaster Core
SMPPro_2303	Showmaster Pro	2.7.3-CAL_369C55f	10.0.0.52:9255	Manager Runtime
MAS-23129999	Medialon Audio Server 2 (UNLICENSED)	2.2.1	10.0.0.150:6543	Medialon Audio Server
ML-LT-001-V6	Manager (UNLICENSED)	7.0.3	10.0.0.253:9255	Manager

This is the quickest way of determining that any device is connected, of the same version, and what its IP address is.

4 Programming Environment

In this section you will discover your main working environment.

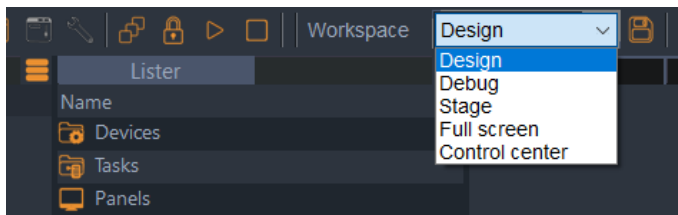
4.1 Workspace

4.1.1 The Workspace

Medialon Manager V7's workspace is composed of areas that display system and device information, graphical user interfaces, or tasks in your project.

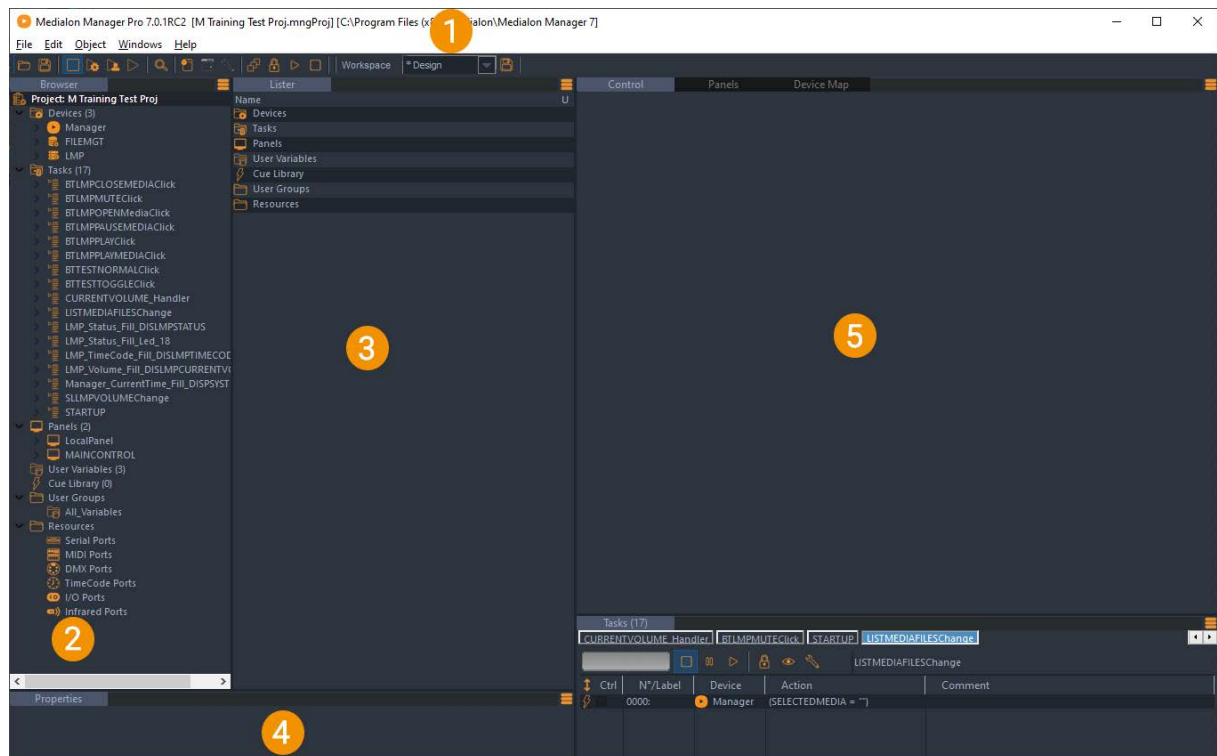
The workspace is defined by all the areas that segment and organize the graphical environment of Medialon Manager, and each area can contain one or several tabs that display the project's information.

Together, the layout of the areas and the distribution of the tabs within the areas create the Workspace. There are predefined workspace configurations from which you can choose:



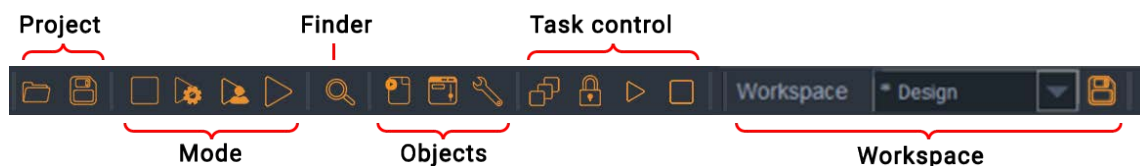
- Design
- Debug
- Stage
- Full Screen
- Control Center

When Medialon Manager starts, the Workspace opens in Design mode as below.



1. The Toolbar
2. The Object Browser tab: contains all your project’s objects, organized by type
3. The Object Lister tab: displays the content of the object selected in the Browser tab
4. The Object Properties tab: the properties of the most recently selected object or cue can be monitored and edited
5. The Working Area could contain:
 - Control tab: where Device Control Panels and Task Control Panels are displayed
 - Device Map tab: a graphical arrangement of Devices with an optional background graphic
 - Panels tab: UserScreen editing tools

4.1.2 Main Toolbar



Medialon Manager’s Main Toolbar gives access to menus and groups of functions.

- Open and Save Project
- Medialon Manager Mode
- Finder

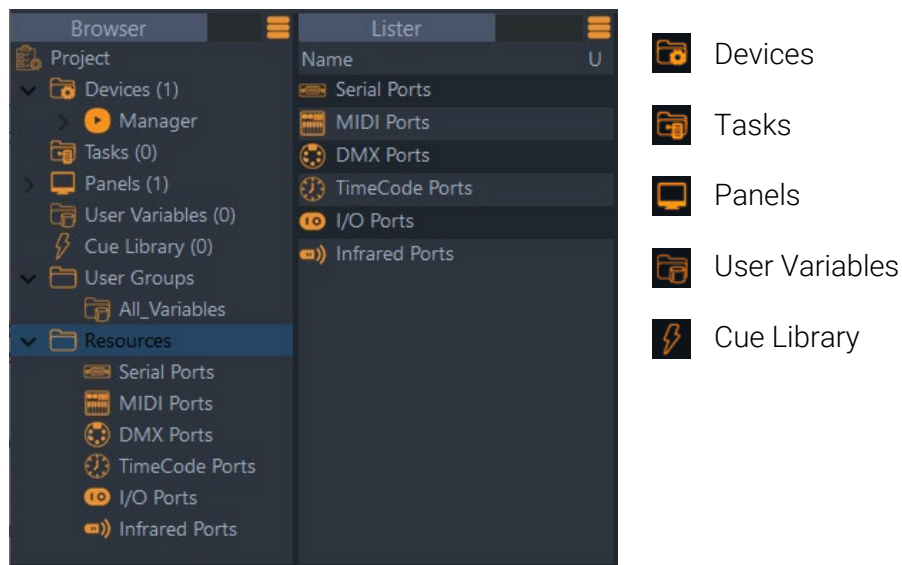
- Objects: New / Control Panel / Setup
- Task control: Stop All / Lock / Start / Stop
- Workspace: Select / Save

4.1.3 Browser Tab

The Browser presents all elements of your project.

Like a tree-type explorer, you can expand or collapse every node. When you select a node, its content is displayed in the Lister area and in the Properties area.

Objects



User Groups Folder

With the User Groups, you can create groups and sub-groups of objects from the same class (devices, variables, tasks or UserScreens), this allows you to organize information that is important for your program. See also [User Groups](#).

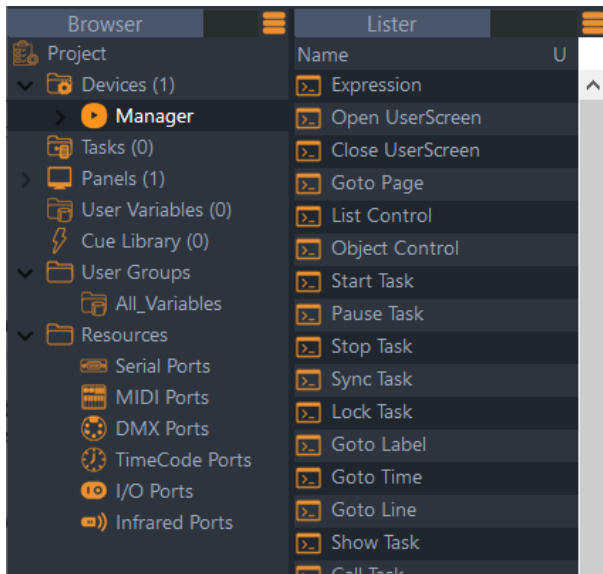
Resources Folder

Resources contains a list of ports available for your project but also allows you to create virtual resources for specific configurations.

4.1.4 Lister Tab

The content of the Lister depends on the Browser selection.

Columns may be sorted by clicking on the column title.



Example: When the Manager device is selected in the Browser, the lister displays the list of Manager commands.

4.1.5 Properties Tab

When an object is selected in the Lister or in an Editor, its properties are displayed in the Properties Tab.



Some properties are read-only and cannot be modified. Only properties with orange corners (top-left and bottom-right) can be changed.

To change a value:

1. Select the field by clicking on it.
2. Enter the new value and press *Enter* to store it.
3. To test a command, set the parameters in the Command Properties and click on the Test Command button to send the command.



Use *Ctrl + Mouse Wheel* to select a read-only property so you can copy it with *Ctrl+C*.

4.1.6 Control Tab

The Control tab is used specifically to display the Control Panels of devices and tasks.

To display an MxM Control Panel inside the Control tab, hit *Shift* + double-click on the device within the Browser tab or inside the Device Map tab.

To display a Control Panel as a floating window outside the Control Center tab, hit *Alt* + double-click on the device within the Browser tab or inside the Device Map tab.

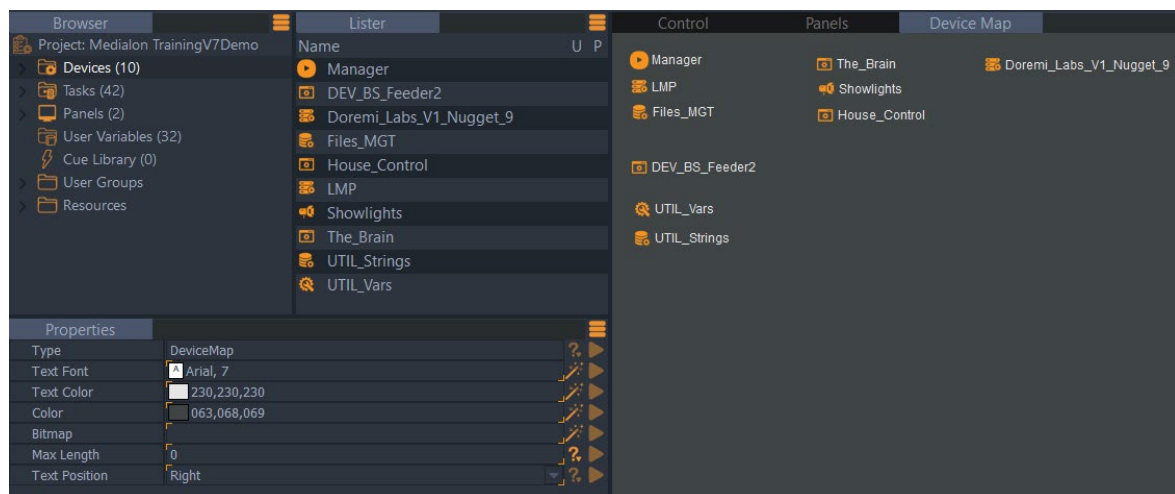
4.1.7 Panels Tab

The Panels tab is where the custom graphical interface is designed and created.

To create a new UserScreen, right-click inside the tab and select the desired options from the menu. For further information please refer to [Panels and UserScreens](#).

4.1.8 Device Map Tab

The Device Map tab is where you create and visually organize your devices. The devices are displayed as icons from which you can access information and setup menus.



To create a new device, right-click inside the tab and select 'New Device'.

To edit an existing device or simply view the device's Help file, right-click on the device icon, and select the desired option. See also [Device Map](#).

4.1.9 Task Tab

The Task tab is where the Task Editor opens. The Task Editor is used to insert and edit actions (cues) inside task.

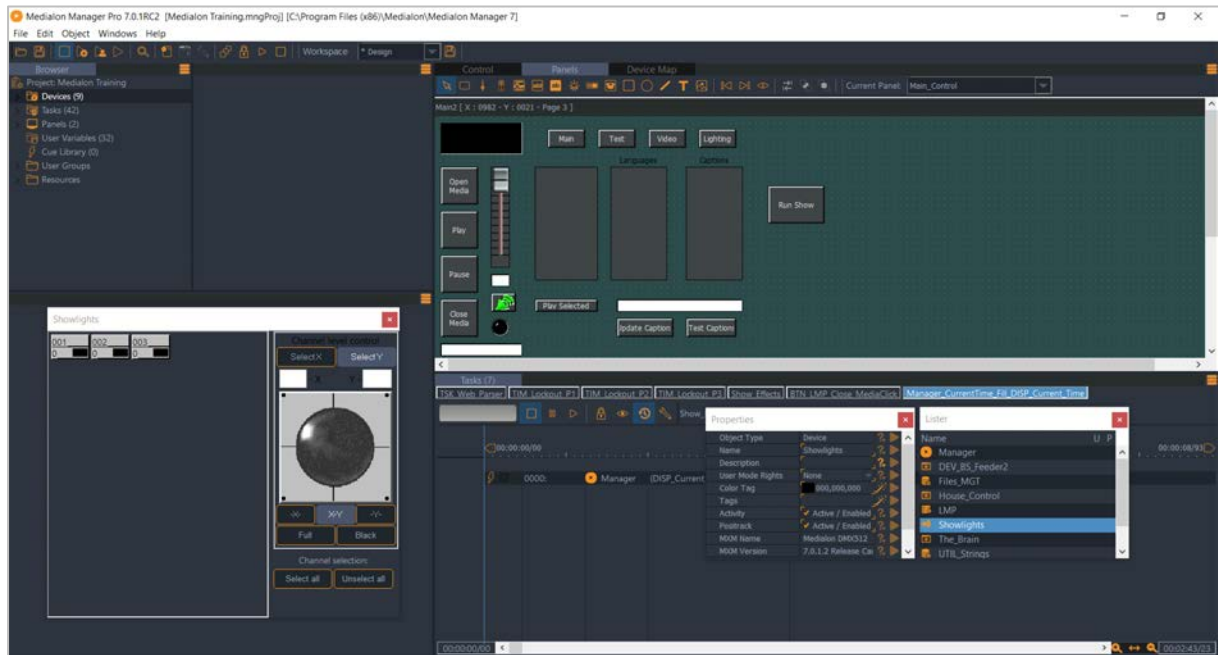
To open a Task Editor, right-click on a task in the Lister and select the desired option. For further information, please refer to [The Task Editor](#).



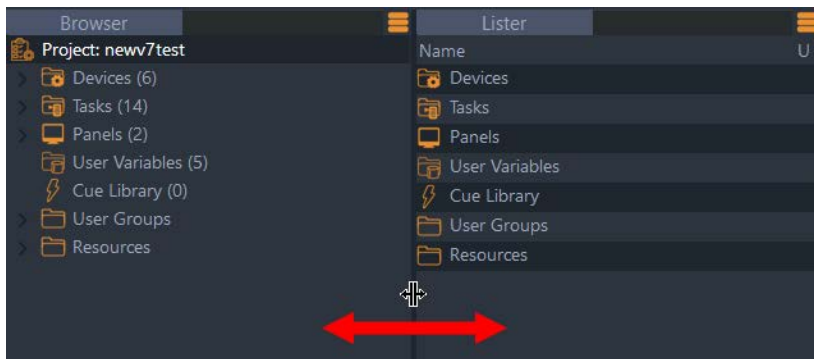
Close all but this – right-click opens menu option that closes all tasks in task window except the task selected.

4.1.10 Customize Your Workspace

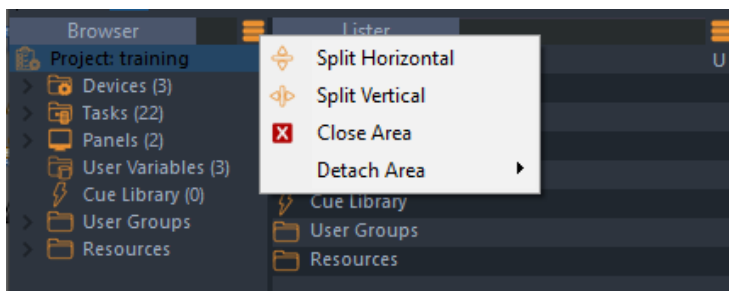
You may choose to customize your workspace to adapt the programming environment to your needs.



Areas can be resized by expanding the edges of the area:

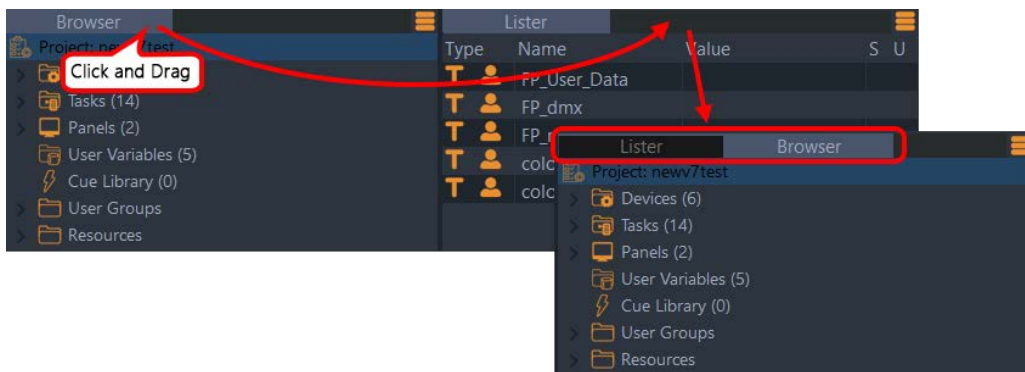


or split vertically and horizontally by clicking on the area management icon.

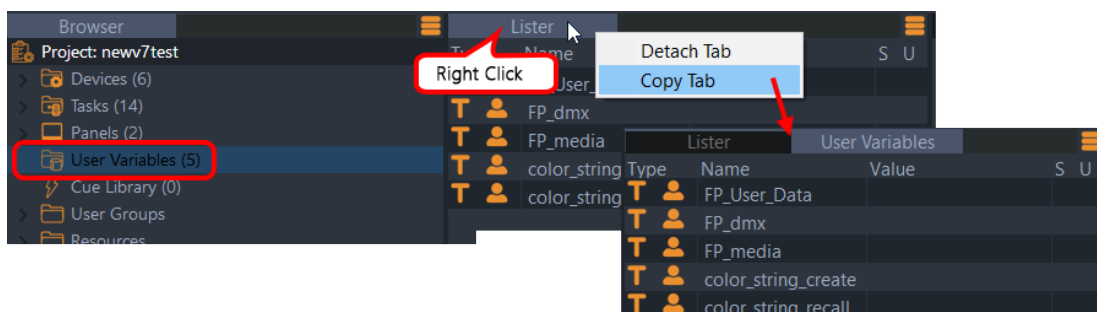


Programming Environment


You can drag and drop tabs among areas to create and organize your own workspace.

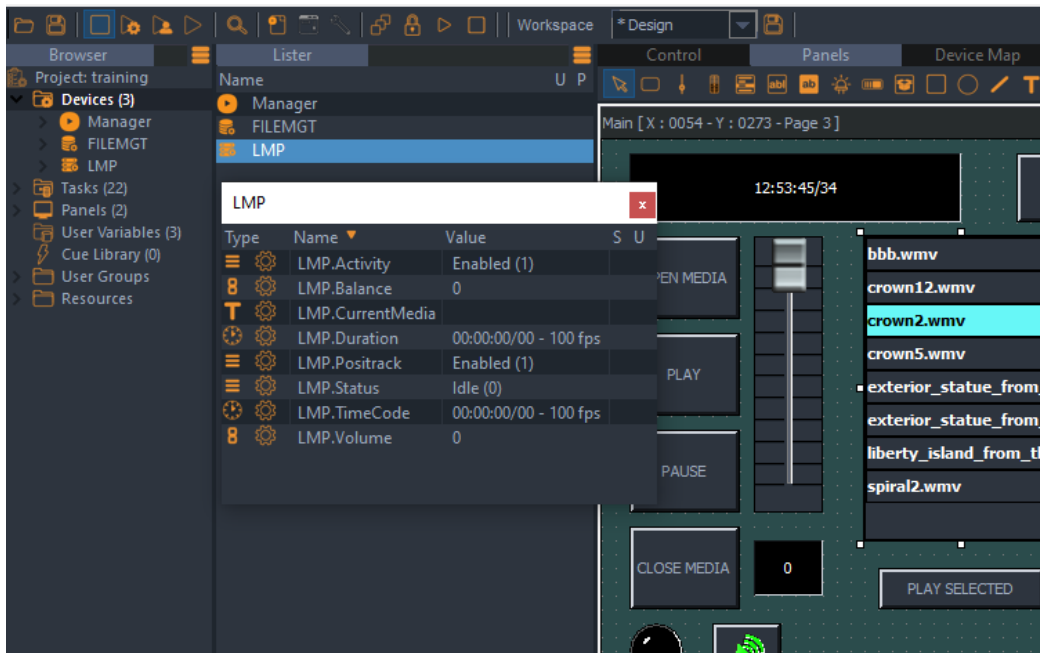


Lister tabs can also be copied. Copied Lister tabs maintain their source selection even if different objects are selected in the Browser afterward (the Lister tab updates, but the copied Lister tab does not).

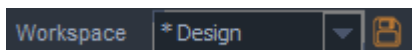



For example: Get a filtered Lister with a list of User Variables commands. The copied lister tab is automatically named based on the contents at the time it was copied.

Tabs can be detached by double-clicking on the tab's title (or right-click the title and select 'Detach Tab'). The tab becomes an independent window that can be moved and resized freely. Click the title bar to reveal the close button:  and restore to the original docked position by closing the panel.



Any changes in the current workspace are marked by an asterisk before the name of the selected workspace.




Save the current workspace by clicking on the Save Current Workspace button .

The new workspace is then available in the list of Workspaces.

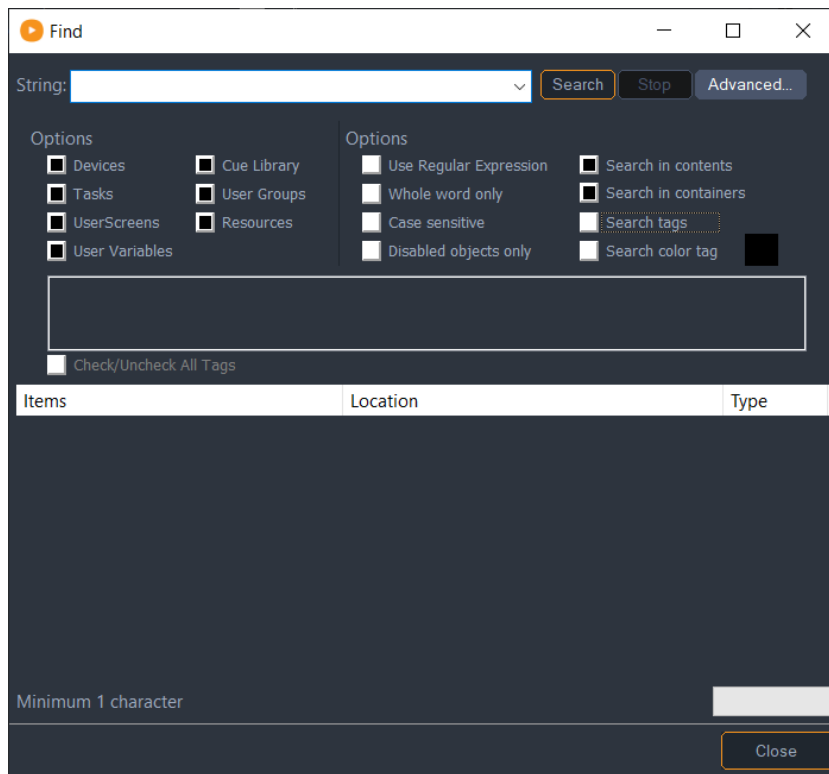
4.2 The Finder

Medialon Manager V7 integrates a search engine that allows you to quickly find the object you are looking for in your project. You will discover that this tool is time saving, especially as your project evolves and becomes bigger and more complex.

 Right-clicking on a device or variable allows access to a wizard to find all references to that device or variable in the current project.

4.2.1 The Finder Window

The Finder can be launched from the Edit menu, by clicking on the Finder icon on the main toolbar, or with keyboard shortcut *Shift+Ctrl+F*. The Find window appears.



String

The String field is where you enter search text, partial or whole.



Click on Advanced to get more options for your search.

Where

You may limit your search to specific areas of your project to avoid unnecessary results.

Devices Option

When this option is selected, the search engine scans Device Names.

Tasks Option

The Search engine scans the tasks to find matching strings. If both the Contents and the Containers options are selected, the engine searches both the task list for names matching the requested string, and the cues inside the tasks for the requested string (Cue Comments are not searched).

UserScreens Option

The search engine scans the UserScreens and the Dynamic Graphical Objects inside them to find matching Screens or Graphical Objects.

User Variables Option

The search engine scans the names and values of user variables to find matching strings.

Cue Library Option

The search engine scans the Cue Library for matching strings.

User Groups Option

The search engine scans for all matched objects in the User Groups.

Resources

The resources are scanned for matching Strings.

Options

Use regular expression

This option enables the Regular Expression Syntax Rules for the search query. Examples:

- `.at` matches any three-character string like hat, cat, or bat
- `[hc]at` matches hat and cat
- `[^b]at` matches all the matched strings from the regular expression '.at' except bat
- `^[hc]at` matches hat and cat but only at the beginning of a line
- `[hc]at$` matches hat and cat but only at the end of a line

If you are not familiar with Regular Expression Syntax, the internet is a good resource for more information.

Whole word only

If you choose to perform the query with the 'Whole word only' option activated, the search engine will only return the exact matches and will not display partial results.

Case sensitive

When the 'Case sensitive' option is enabled, only exact matches of upper and lower case characters will be returned in the results window. Combined with the 'Whole word only' option, only identical results will be returned.

Search in contents

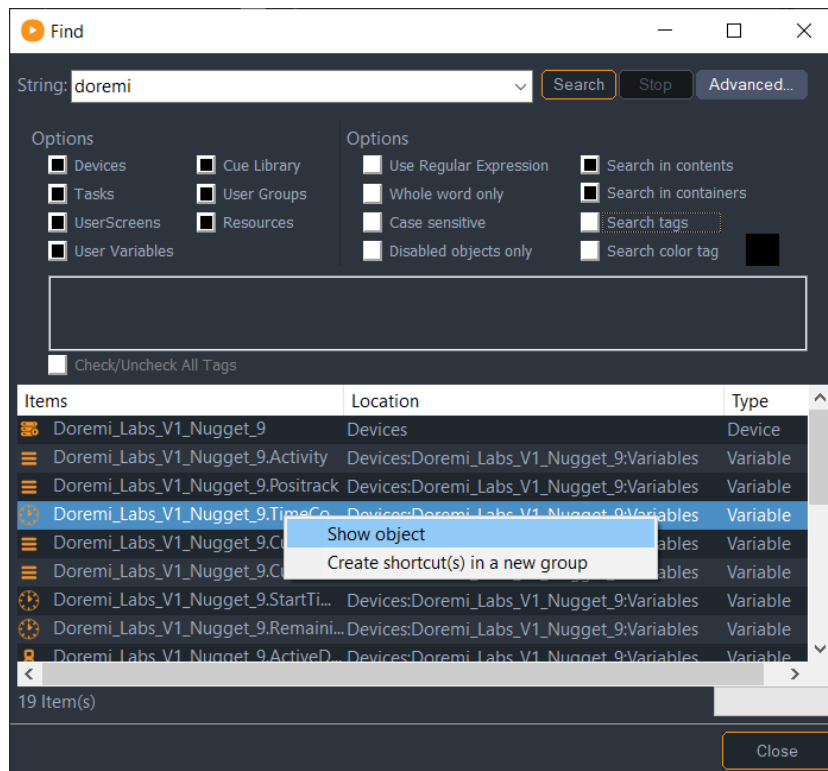
When the Search engine performs its scan, it has the option to search the contents of certain objects. For instance, if you wish to find a cue inside a task, you must enable this function as well in the selecting 'Tasks' in the 'Where' menu. The engine will search the 'Content' of the tasks for the desired string.

Search in containers

'Searching in containers' enables the search to take place in the objects listed in the 'where' menu. For instance, if you are searching for a task, you must enable this option to scan the task names.

The Results Section

All results of the search query are displayed in the Results list.



Right-clicking or double-clicking on the desired object opens a contextual menu allowing you to force the Browser and Lister to display the selected object. You may also choose to create a User Group from the selected result(s). Multiple selection is available using the *Ctrl* and/or *Shift* keys.

5 Show Programming

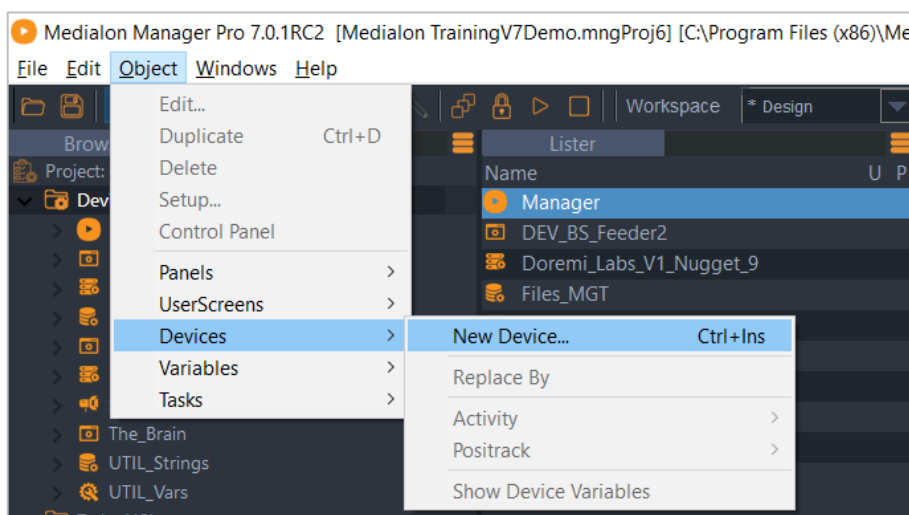
In this section you will learn show programming from the basic single show structure to complex multiple show structures using timelines.

5.1 Devices

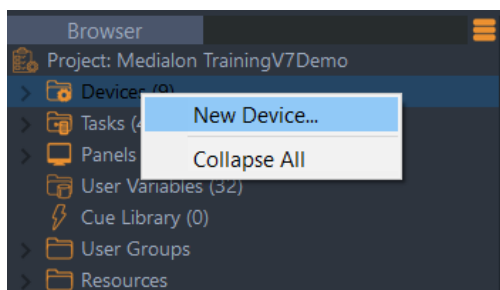
Devices are the machines, programs, or services that Medialon Manager controls with the MxMs and the hardware interfaces connected to it. An MxM can host several devices. For example, you may create as many Doremi devices as you wish using the MxM Doremi Labs V1/Nugget.

5.1.1 Creating a Device

1. Select in the menu bar **Object > Devices > New Device**, type *Ctrl+Insert*, or right-click in the Device Map to open the contextual menu and choose **New Device**.



Or select the **Devices** item in the **Browser** and right-click to choose **New Device**.

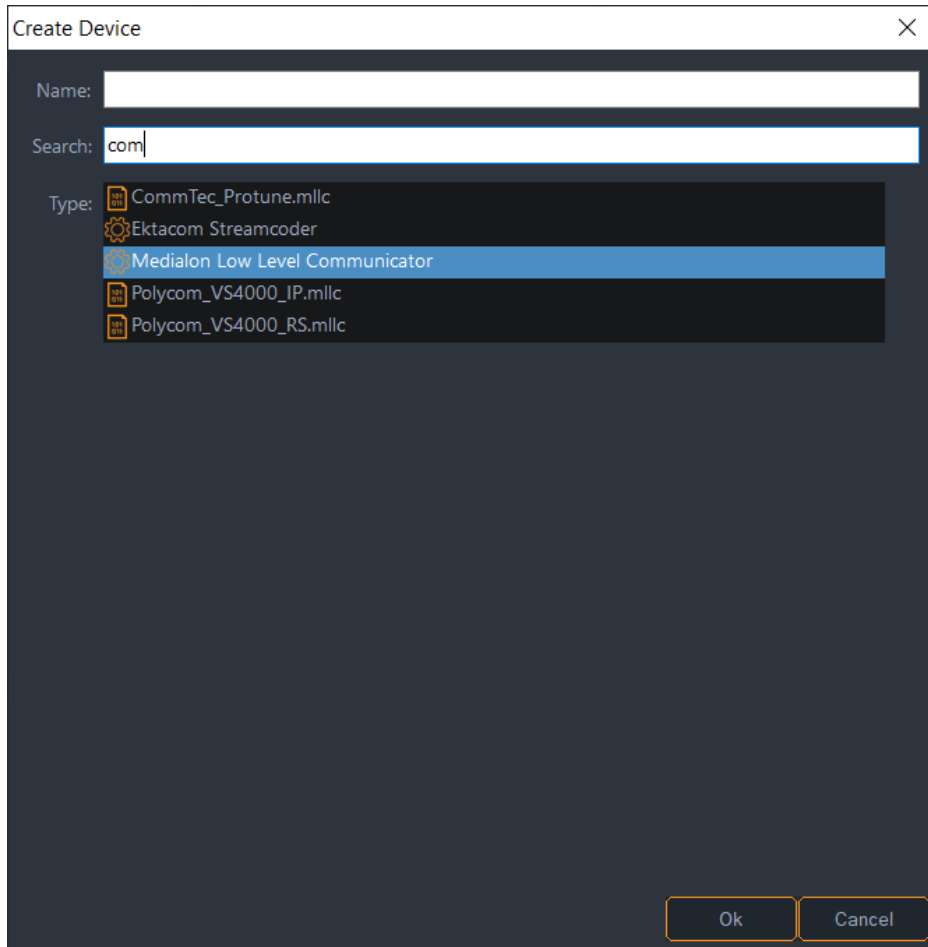


The **Create Device** window opens.

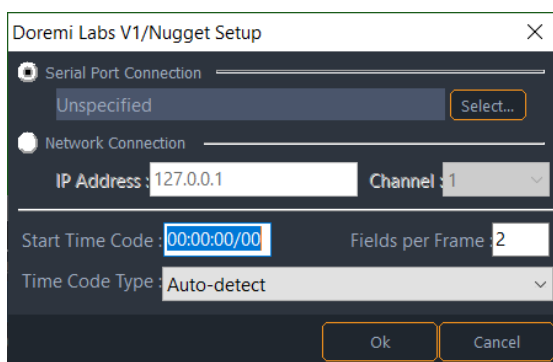
2. Select the type of device you want to create from the list and give the device a name in the **Name** field. The list of device types contains MxMs, Low Level Communicator drivers, and Scripts that work with MxM Medialon Script. (Look for 'Medialon Low Level

Show Programming

Communicator', 'Medialon Script'.) Start typing text into the Search field to filter the driver list.

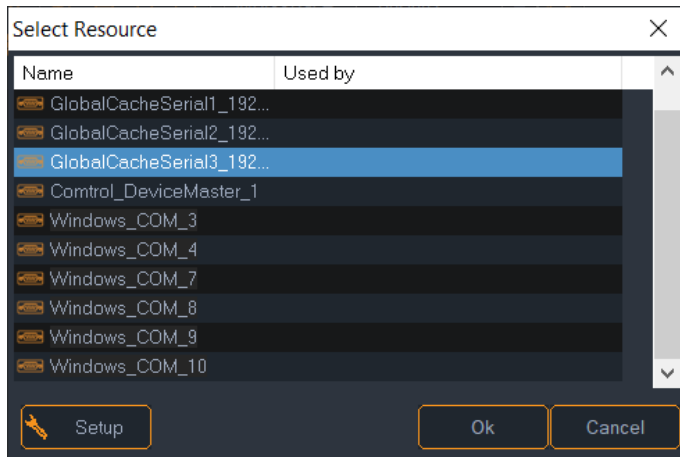


3. In the Device Setup dialog box, the required parameters and configuration can be set. The contents of the Device Setup dialog box are different for each type of device. Please refer to MxM help files for correct device setup configurations.



Show Programming

- Click 'Select' to open the Resource Selection Box.



- Select an available resource and click 'OK'. If the resource is not yet defined, you can keep the resource Unspecified or use the 'Setup' button to create a new resource.
- Once you have finished configuring the device, click 'OK'. The new device is created and appears in the Browser tab and Device Map tab.

5.1.2 Device Properties

Select a Device in the Browser tab or in the Device Map tab; its properties are displayed in the Properties tab.

Properties	
Object Type	Device
Name	Doremi_Labs_V1_Nugget_9
Description	My description
User Mode Rights	None
Color Tag	000,000,000
Tags	
Activity	Active / Enabled
Positrack	Active / Enabled
MXM Name	Doremi Labs V1/Nugget
MXM Version	7.0.1.2 Release Candidate
MXM File Name	mxmDoremiLabsV1.mxm
MXM Manufacturer	Medialon

Read-only MxM information such as MxM file name and version used by this Device is available here.

You can also rename the Device. Do not forget to validate your changes with the *Enter* key. The Description field allows extra (optional) information.

User Mode Rights defines if this device object is visible in User Mode.

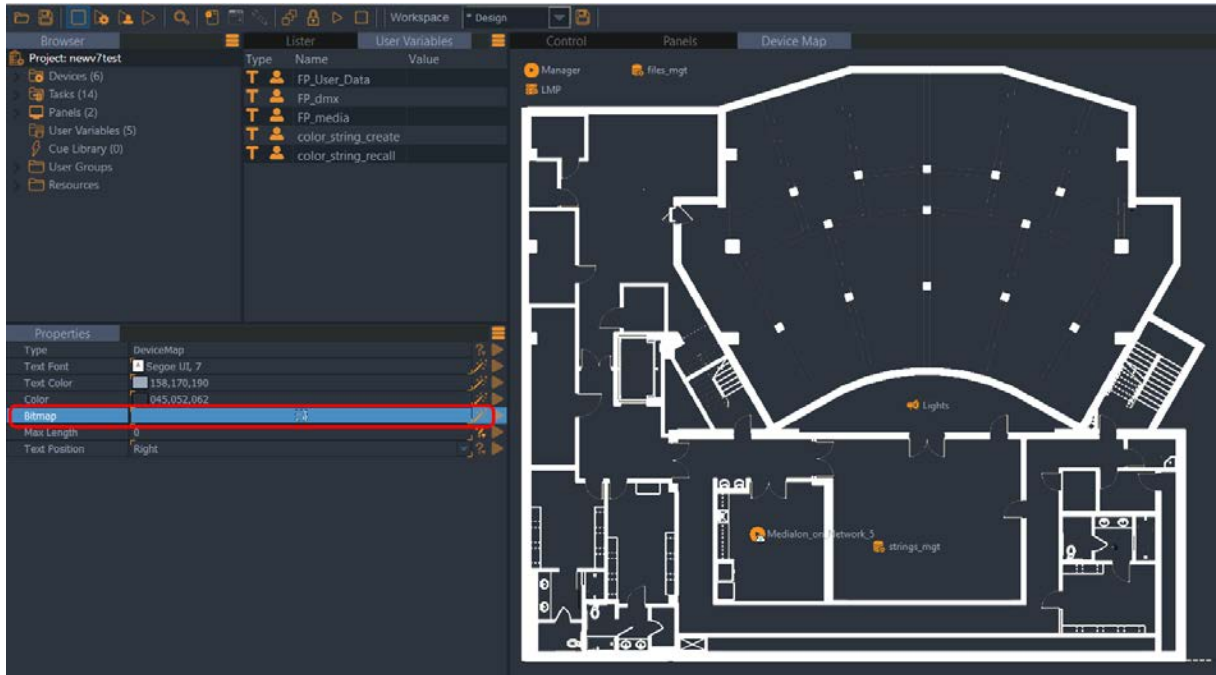
Device Activity can be enabled or disabled. If Device Activity is disabled, connection with the real device is not opened and no commands generated by Medialon Manager are sent to the real world device.

Device Positrack can be disabled, See also [Positrack](#).

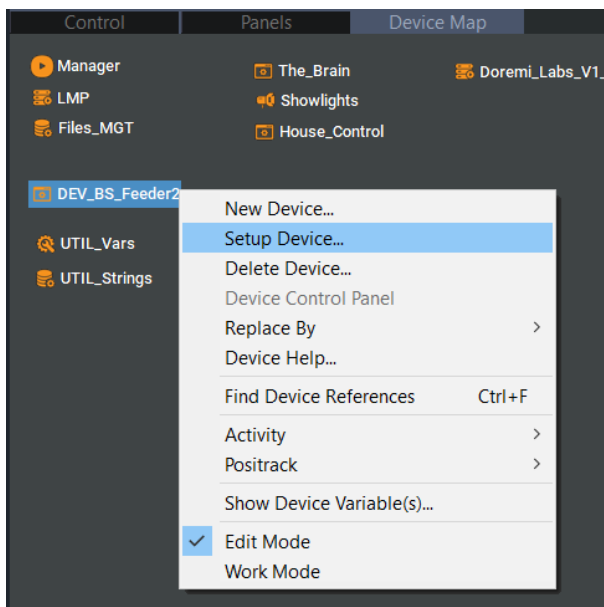
5.1.3 Device Map

The Device map is an alternative way to present the device list contained in your project.

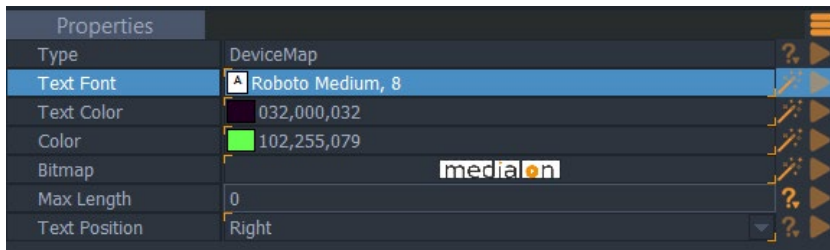
The Device Map can also be used as a graphic representation of your system, using a floor layout image as a background map. Controls and devices can then be positioned on the plan to represent their physical position or working area.




The right-click menu presents the same set of commands as the Device Lister. The Device Map can be used in Edit Mode where you can move Device icons around on the map or in Work Mode where Device icon positions are locked.



Device Map Properties



 Use the wizard button to open a wizard to change properties value.

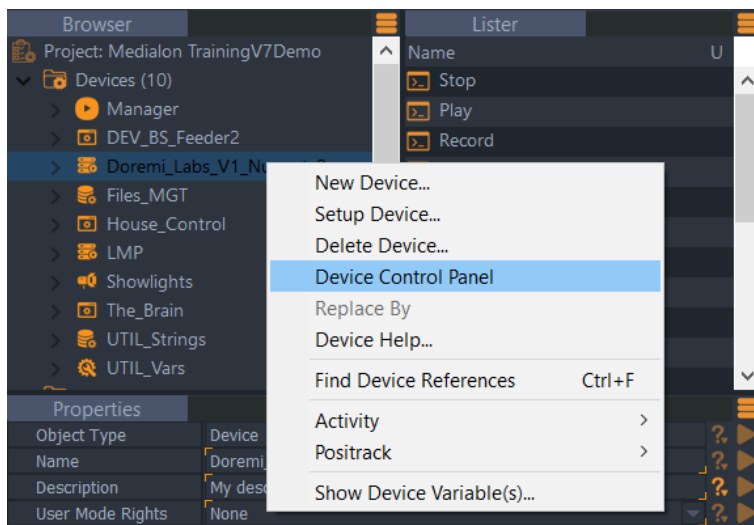
- *Text Font*: Change font, size, and decoration of device name in the Map. See also [Font Selector](#).
- *Text Color*: Use the Color selector to change the text color.
- *Color*: Use the Color selector to change the background color of the Map. See also [Color Selector](#).
- *Bitmap*: Use the Image selector to add an image as background of the map. See also [Image Selector](#). Note that .png images can be used if you require transparency.
- *Text Position*: Change the position of the device name around the device icon.

5.1.4 Testing Devices

After creating a Device, Medialon Manager V7 allows the user to quickly test the device settings and connections without any programming.

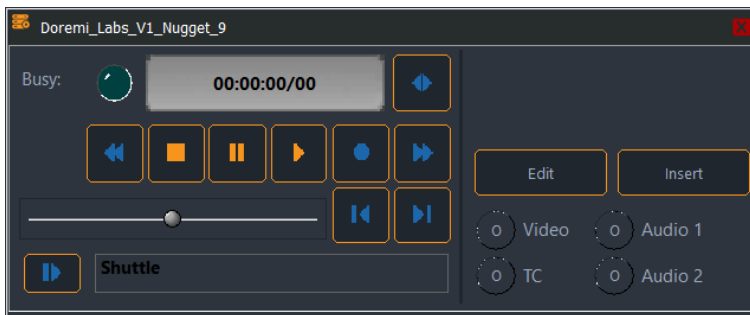
Testing Device Control Panels

1. Press the F2 key on your keyboard to switch Medialon Manager to Debug Mode.
2. Select the device in the Browser tab (or in the Device Map tab): use the right-click menu to open the Device Control Panel in the Control Center.



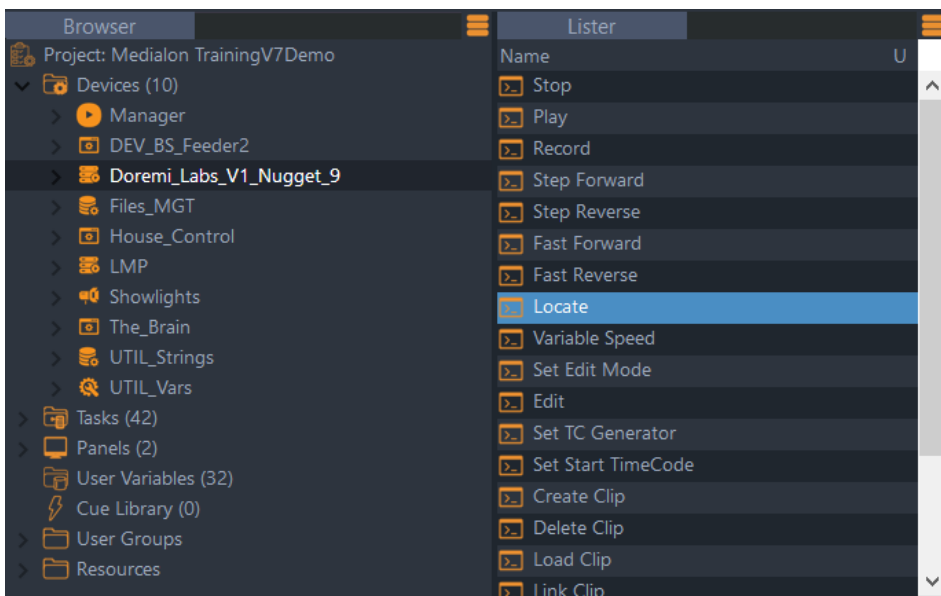
Show Programming

3. Check that the device is online.
4. Control the real world device with the buttons in the device control panel.



Testing Commands

1. Select a Device in the Object Browser; the full list of available commands appears in the Object Lister.

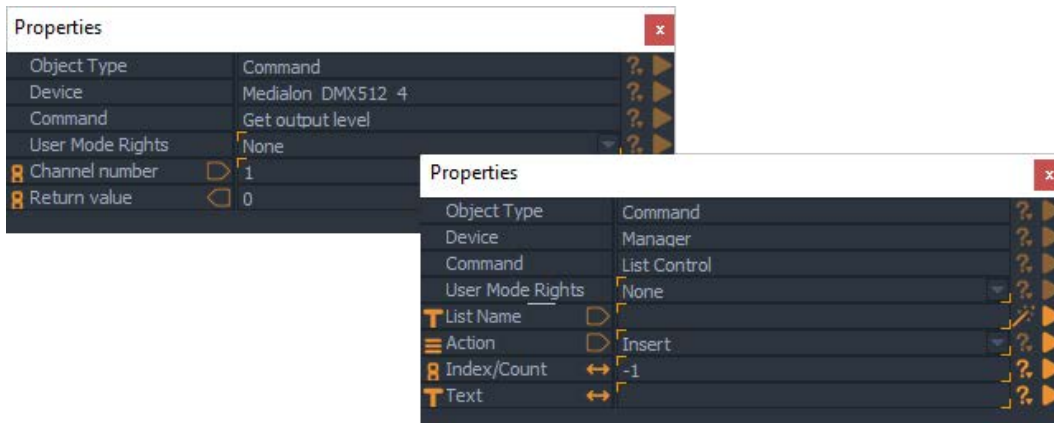


2. Select a Command in the Lister.
3. Enter parameter values, if needed, in Object Properties.
4. Click on the *Test Command* button in Object Properties.



Arrows indicate the direction of properties. Properties could be Output, Input or Both. Input (right arrow) indicates that the parameter is given to the command (used by the command during execution), Output (left arrow) indicates that information will be returned by the command; use a variable on Output type Properties to update the named variable with the information returned by the command (after the command is executed).

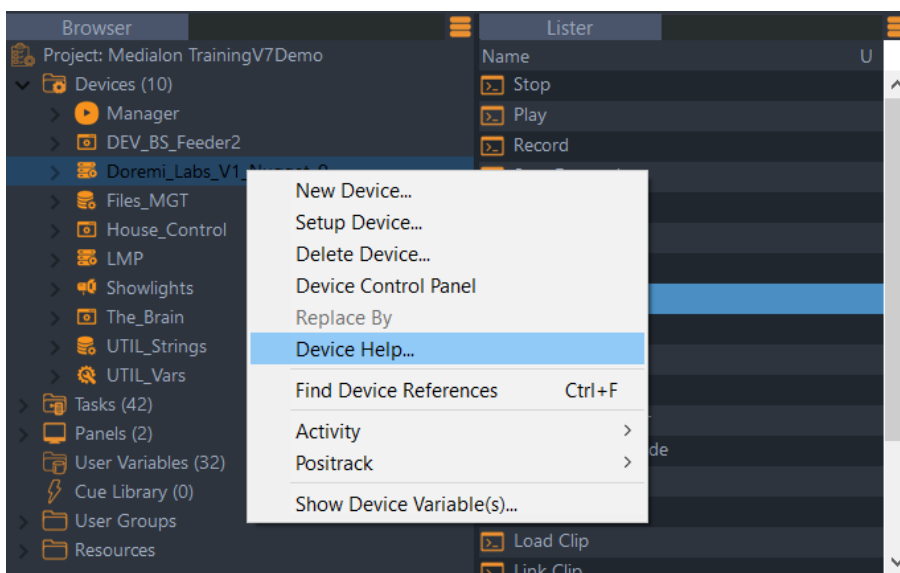
Show Programming



5.1.5 Device Help

Medialon Manager V7 provides a Help file for each Device.

Select the device in the Browser tab (or in the Device Map tab): use the right-click menu to open Device Help.



Each MxM is provided with a complete Help file.

medialon

MXM List (Alphabetical) MXM List (Category)

Medialon MxMs' Help

Name: DoremiLabs V1 & Nugget
Version: 7.0.0
Limitation In:
Device Brand: Doremi Labs
Positrack Compatible: Yes
Resources type: UDP/IP Network, Serial

Compatible hardware interfaces - available resource modules (MRC):
[Serial] AMX
[Serial] Control Device Master
[Serial] Global Cache
[Serial] Sealevel SeaLINK
[Serial] Showmaster LE
[Serial] Showmaster Serial
[Serial] Windows COM

> Overview | > Installation (MXM) | > Creation (Device) | > Commands (List Of) | > Variables (List Of) | > Support

Overview



Doremi Labs offers a range of MJPEG ,MPEG and uncompressed high quality video players and recorders. This MxM allows the control of Doremi Labs V1 or Doremi Labs Nugget machines. This type of video recorder/player offers an interesting alternative to the standard analog video recorders. The connection between the MXM and the device could be either a RS422 connection (Sony 9 Pin connector) or a Network connection (UDP/IP).

In addition to the standard VTR commands, this MxM provides Clips management function. Clips are named Segments in the Doremi V1 litterature (which is more closed to what a clip is, inside a V1) for single file machines and are named Files for multi file machines.

Single file machines are seems as a 24h virtual tape machine like a VTR whereas multiple files machines are seems as video file servers (multiple 24h virtual tape). In this last case, each file has a name (Clip name) composed by a maximum of 8 characters.

The Help document contains:

- Overview of the real Device
- Information about MxM installation, like if a third-party driver or software is needed
- Device Creation and Setup
- List of Commands with comments and parameters
- List of variables with the description of available feedback information
- Release Notes of the MxM



All MxMs are available from 7thSense (see <https://portal.7thsense.one/medialon-help/>).

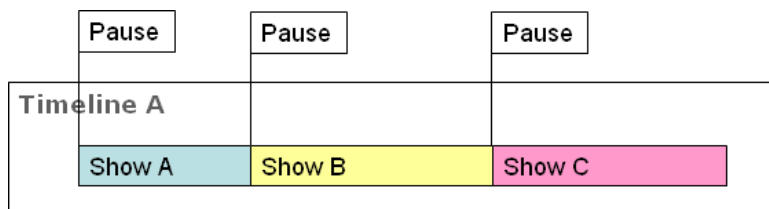
5.2 Time Based Tasks

5.2.1 Show Structure

When you create a time based show with Medialon Manager, many structures can be used with single or multiple timelines. This must be considered in advance and is dependent upon the operational requirements of the show. Among the many show structures that Medialon Manager V7 allows you to build, here are two basic examples typically used when programming.

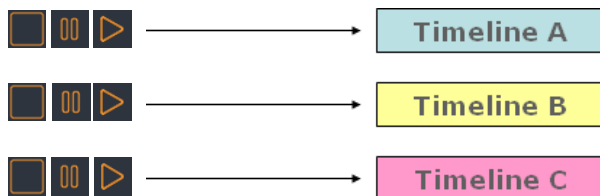
Single Timeline

The main show is built on one single timeline. Segments of the show can be played consecutively one after the other.



Multiple Timelines

The main show is subdivided into several timelines.



Timelines can be played individually or simultaneously by synchronizing them with the same timecode (internal or external).

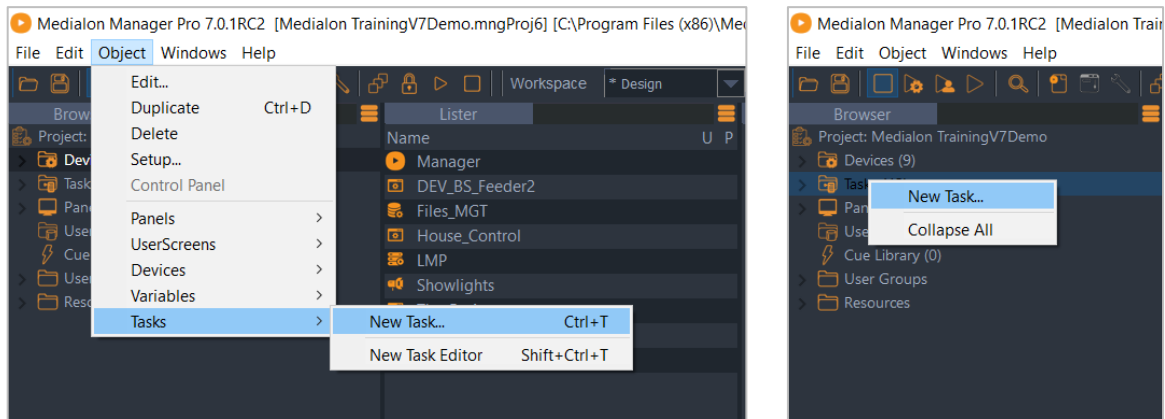
5.2.2 Creating a Time Based Task

A time based task is the simplest type of task that Medialon Manager provides. A time based task is made of tracks that contain cues. The time based task's current time position is the reference time (which may be either internal or external). Cues are triggered when the reference time reaches the cue's time position.

Creating a time based task

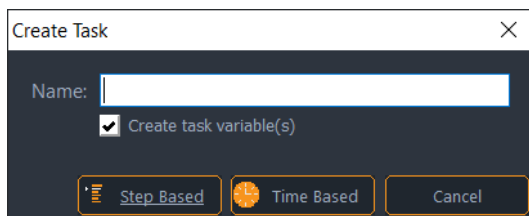
1. Select the 'Object/Task/New Task...' menu item.

Show Programming



Optionally, you can select the Tasks item in the Browser tab and right-click to open the contextual menu and choose New Task.

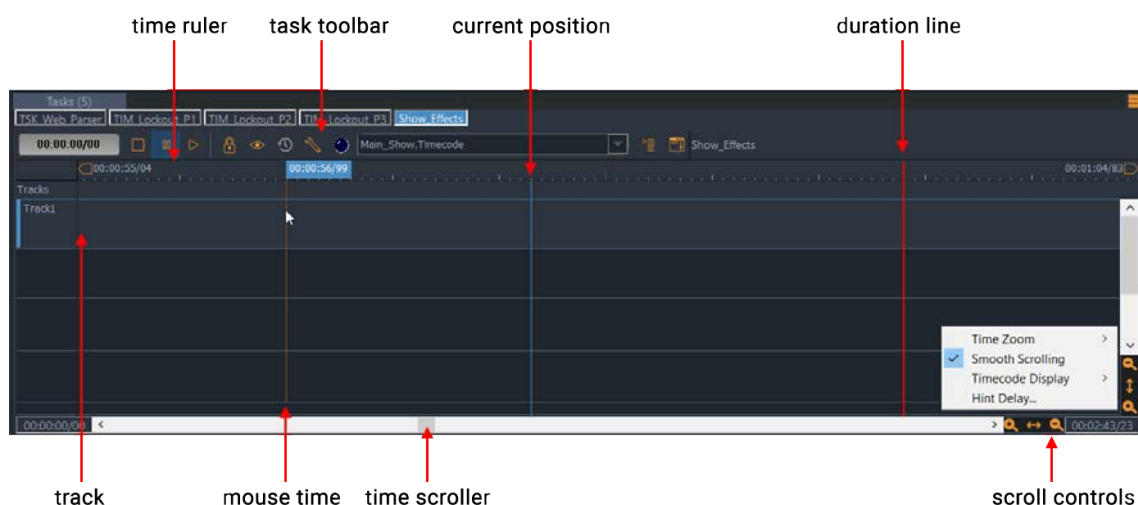
2. The Create Task dialog box is displayed:



3. Type a name for the task.
4. You can create variables that will reflect various statuses of the task by checking the Create task variables box.
5. Click the 'Time Based' button.
6. Medialon Manager creates the time based task and opens it in a new Task Editor tab.

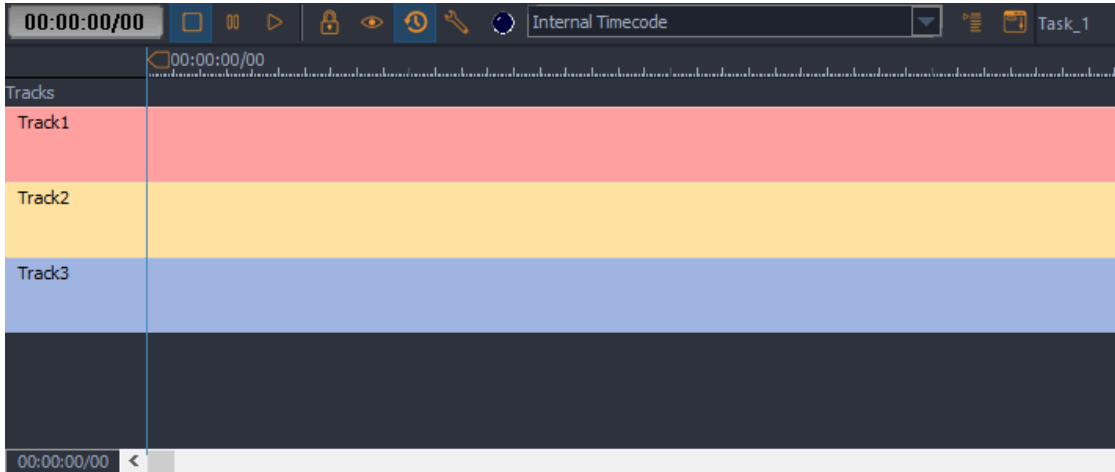
The Task Editor

The Task Editor shows the task's contents and is where you program your time based task.



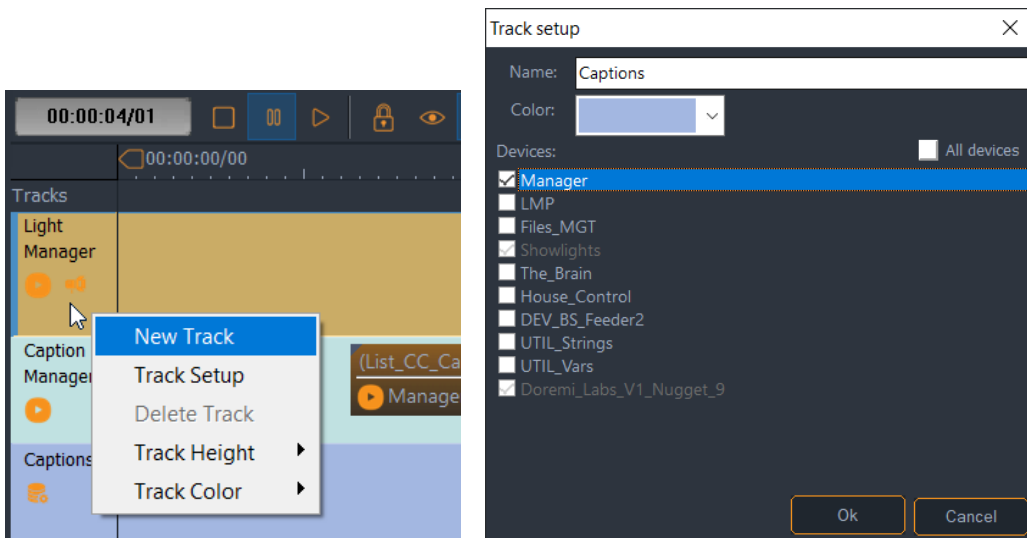
5.2.3 Track Management

Tracks are a way to organize your work: you may want to put all the cues for all devices on the same track, or you may want to use different tracks for each device.



You can create and use up to 99 tracks in a timeline.

Right-clicking the mouse on the left part of the track displays a contextual menu that allows you to customize the track's aspects:



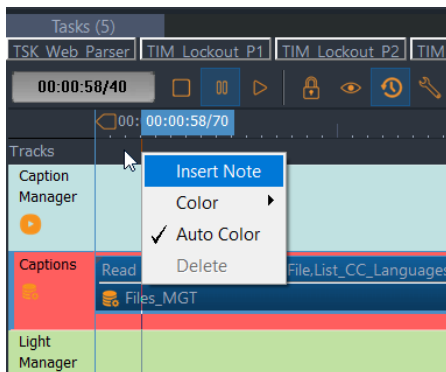
You can change the color, the height, or the Setup of the track. You can create new tracks or delete existing tracks. You may also decide to allow only specific devices' commands on the track.

Track Note

Track notes are used to mark sections of the timeline graphically. They are static and do not send information to the system but can be very useful during programming to rapidly identify sections of your show.

Show Programming

To add a note to your timeline, right-click on the section between the Time ruler and the first track and choose the 'Insert note' option. Once the note has been added, simply double-click on it to edit the note's text field. You may choose to have colors of the note be automatically managed by the system.



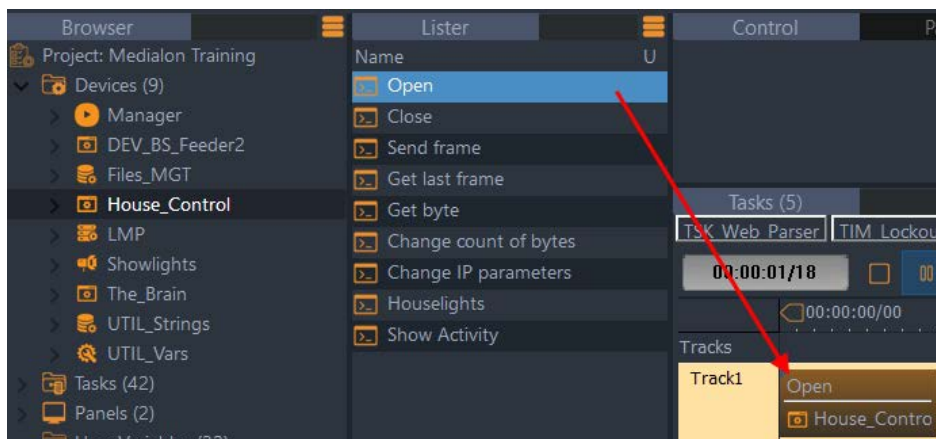
Once the note has been inserted, you can change the color or delete it by right-clicking on the selected note.

Notes may be resized by clicking and dragging the right edge of the note and moved by clicking and dragging the left edge.

5.2.4 Creating Cues

Method A

1. Select the desired device in the Browser tab to display the commands in the Lister tab.
2. Select a command in the Lister tab.
3. 'Drag and Drop' this command where you want to insert the cue in the task's timeline:

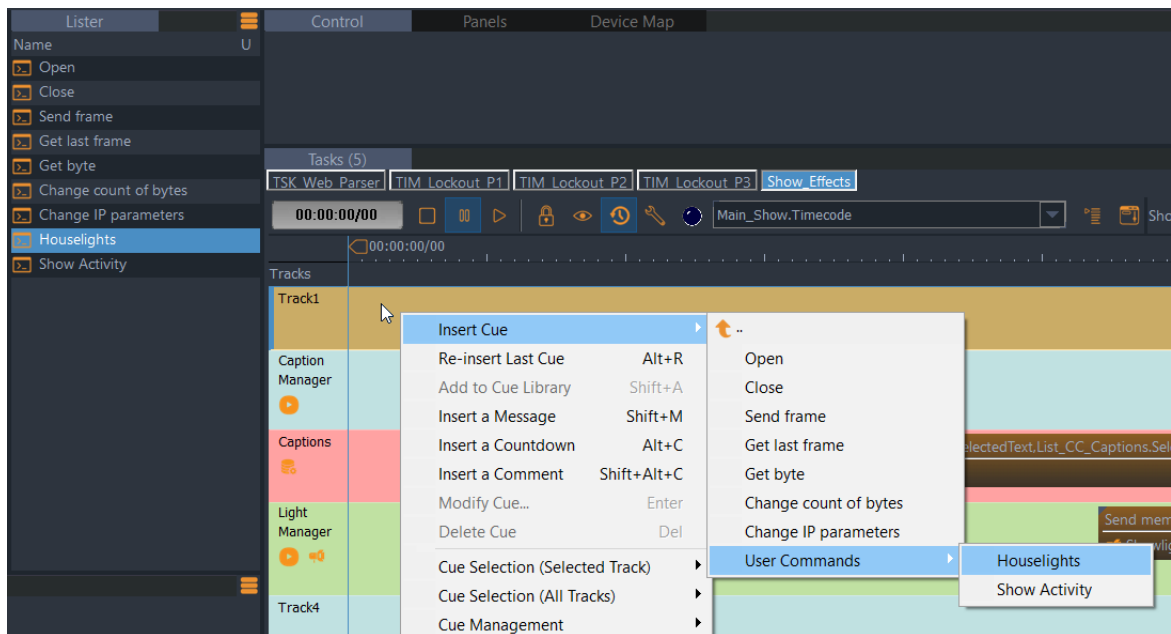


Method B

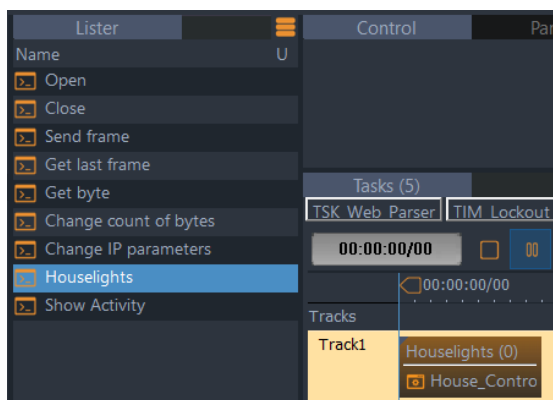
1. Move the mouse pointer to the desired cue position on the timeline.
2. Right-click and select 'Insert Cue'.

Show Programming

- Then browse to the desired device command and select it.




- Select 'Houselights', for instance.



- The desired command cue is inserted in the timeline.

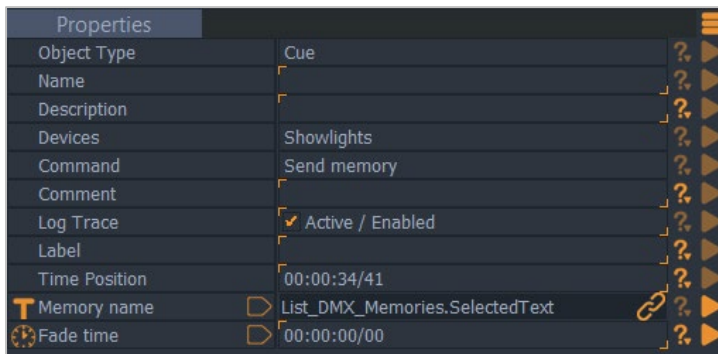
Testing

- To test the cues, press the *F2* button on the keyboard to switch to Debug Mode, then click on the Play button of the task toolbar: 

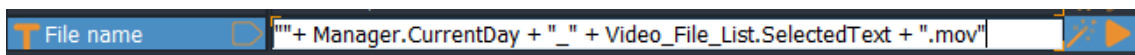
Cue Properties

Select a cue in the task to display its properties in the Properties tab:

Show Programming



i Properties of cues can include Expressions, eliminating the need for extra single-use User variables and shortens your tasks by moving the calculations into the cues to which they are related to.



- *Devices*: Name of the device or devices affected by the cue
- *Command*: Name of the selected command
- *Comment*: You can add a comment about this cue or command
- *Label*: Each cue is identified by its Time Position. A 'Label' may also be added to it. The Label is associated with the cue and thus is not changed if the Time Position shifts.
- *Time Position*: Time Position of the cue on the timeline
- *Command Parameters*: dependent on the command

Inserting a Message Cue

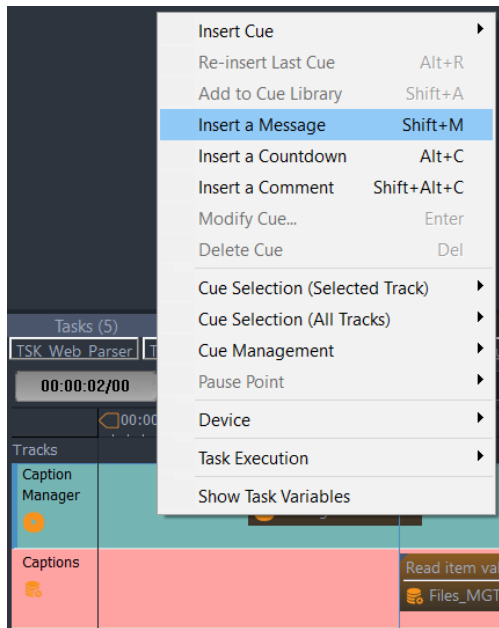
Message cues are cues that have no effect on external devices but are used to dynamically document the show and track information in task variables while the timeline is running.

Message cues display their messages in the Task Control Panel and are also available as a Message variable if you chose to create task variables when the task was created.

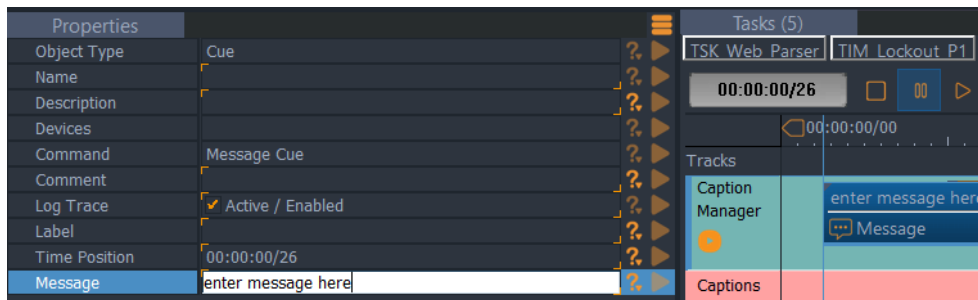
To insert a Message cue:

- Type *Shift+M* or select the 'Insert a Message' menu item, or right-click inside the track.

Show Programming



- Enter the Message text and hit *Enter*.

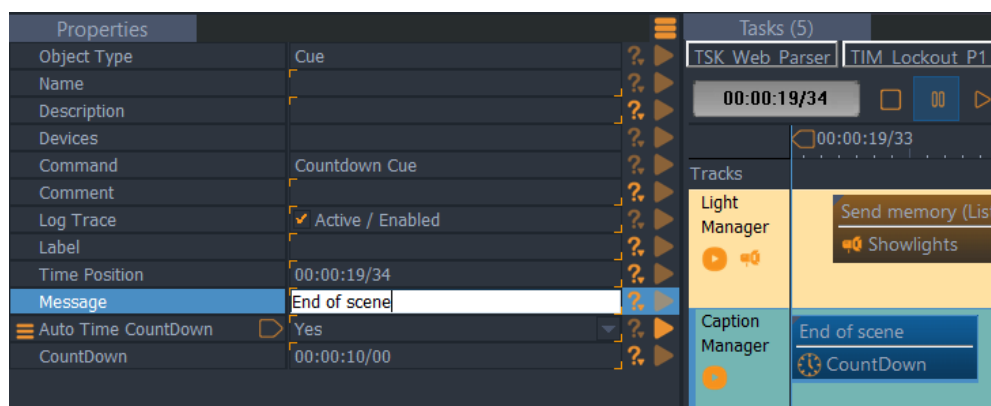


Inserting a Countdown Cue

Countdown cues are similar to Message cues but with advanced functionality. The Task Control Panel displays the remaining time before the next Countdown cue with an associated message.

To insert a Countdown cue:

Type *Alt+C* or select the 'Insert Countdown Cue' menu item, or right-click inside the track. Type a message text and hit *Enter*.



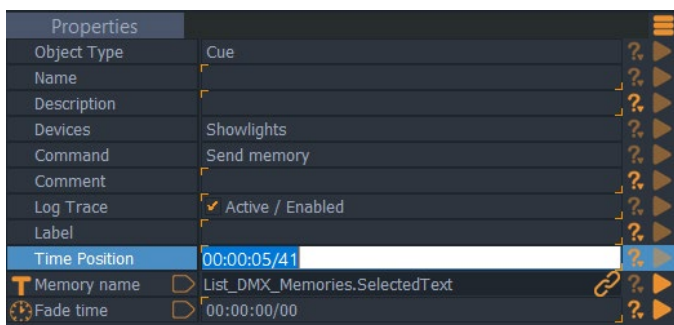
Show Programming

- Auto Time Countdown means that the countdown will start from the previous countdown cue, or from the beginning of the timeline if there are no previous countdown cues.
- When 'No' is selected, the countdown will count either from the Countdown Time (X time from the countdown cue's position), or from the end of the last countdown cue if it is within the Countdown Time period.

Changing Cue Properties


Select a cue to display its properties to modify Comment, Label and Time Position of the cue.

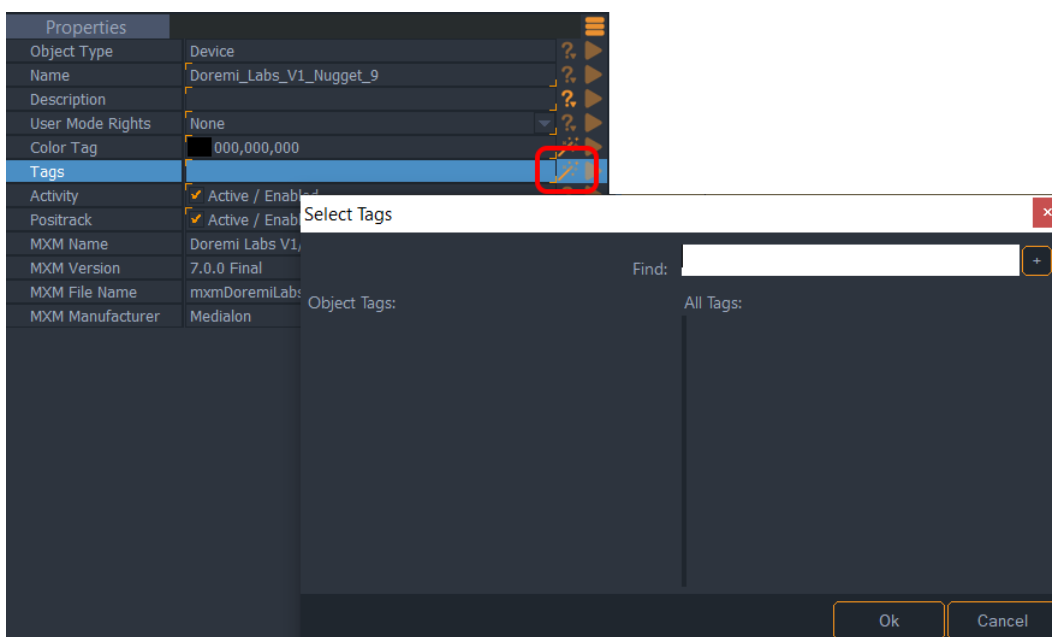
A cue's properties are automatically displayed in the Object Properties for editing when you create them or select them. Only properties with orange corners (top-left, bottom-right) can be changed.



i Remember to validate your changes with the Enter key.

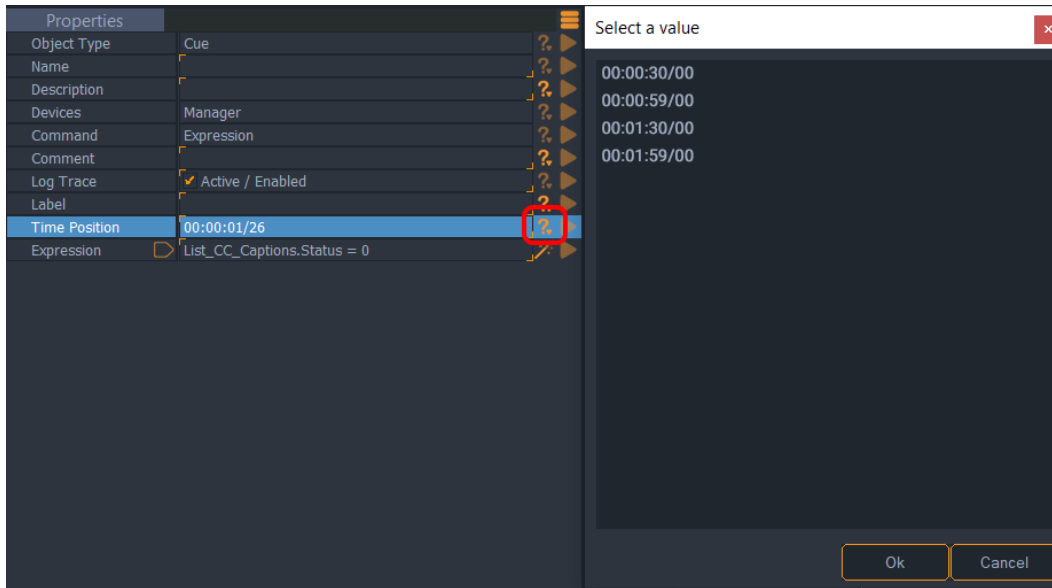
The number and type of parameters in the Properties tab depends on the cue type (device command, message, countdown, etc.).

Some parameters have a wizard to help you choose a value. To open the wizard click on the Magic wand icon  on the right side of the parameter.



Show Programming

If no wizard is available, the question mark button allows you to select one of the last values you already manually entered.



5.2.5 Selecting Cues

There is often the need to select cues before using cue related commands. That is why the Task Editor provides a variety of ways to select cues. Selected cues are highlighted in orange.

To select cues:

- Click on the cue to select one cue.
- Press the *Ctrl* key and click on another cue to add it to the selection.
- Press the *Shift* key and click on a cue to add of the cues which are between the last selected cue and this cue.
- Type *Ctrl+A* to select all the cues in the task if no track is selected, or all the cues in the current track, if a track is selected.
- Drag the mouse to draw a selection box over the cues you want to select.

To deselect cues:

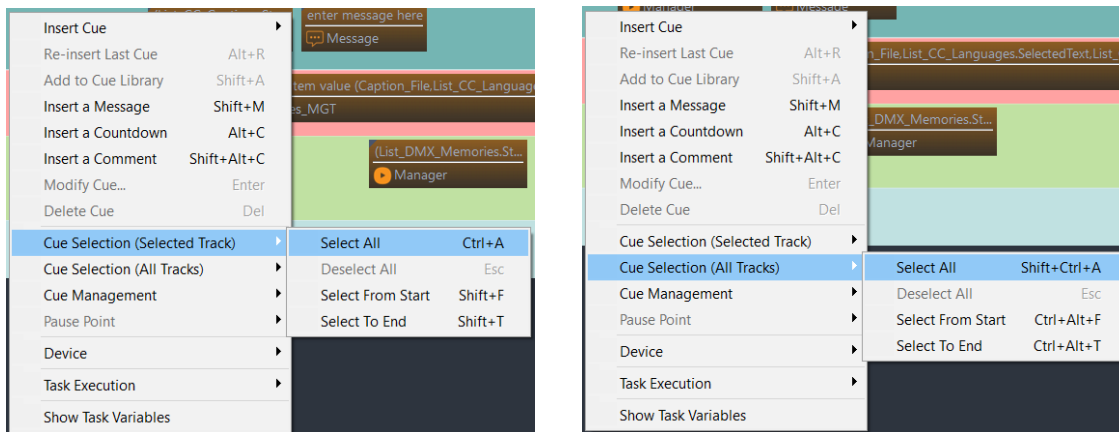
- Click on an empty part of the task editor to deselect all the cues.
- Press the *Ctrl* key and click on a selected cue to remove it from the selection.
- Type *Esc* to deselect all cues.

The 'Cue Selection' menus

The cue selection can be done inside the active track or in all tracks of the timeline. In both cases, the list of commands is the same.

Show Programming

These menus allow more sophisticated types of cue selection.



5.2.6 Managing Cues

Cut, Paste and Delete

Cut and paste commands are supported with familiar keyboard shortcuts:

- Type the Delete key to delete the selected cues
- Type Ctrl+C to copy the selected cues
- Type Ctrl+X to cut the selected cues
- Type Ctrl+V to paste cues to the current position of the timeline. Pasted cues maintain their relative offset from one another, just as they were Copied or Cut.

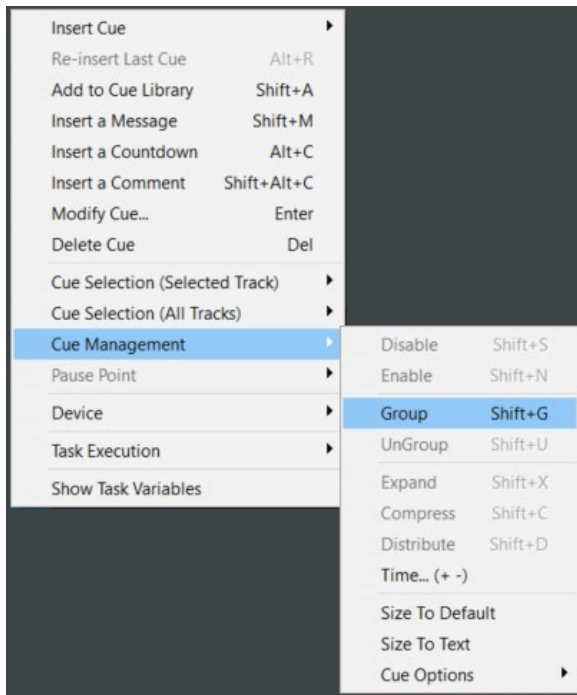
Moving Cues

There are several handy keyboard shortcuts to precisely move selected cues:

- Hold the *Alt* key down and use *arrow* keys to move the selected cues by 1/10th second increments. Use *Ctrl+Alt* and *arrow* keys to move them by 1/100th second increments.
- Type the *+* key and enter a time to move the selected cues forward.
- Type the *-* key and enter a time to move the selected cues backward.

The 'Cue Management' menu

The Task Editor also provides various commands in the 'Cue Management' submenu to automatically change relationships between cues.



Cue Disable/Enable

- Disable / Enable cues: Disabled cues are not executed during task execution.

To disable a selected cue:

- Type Shift+S
- If the Flash Control feature is activated, click on the Enable/Disable Flash Control Button
- Choose the 'Cues Management/Disable Cues' menu item

To enable a selected cue:

- Type Shift+N
- If the Flash Control feature is activated, click on the Enable/Disable Flash Control Button
- Choose the 'Cues Management/Enable Cues' menu item

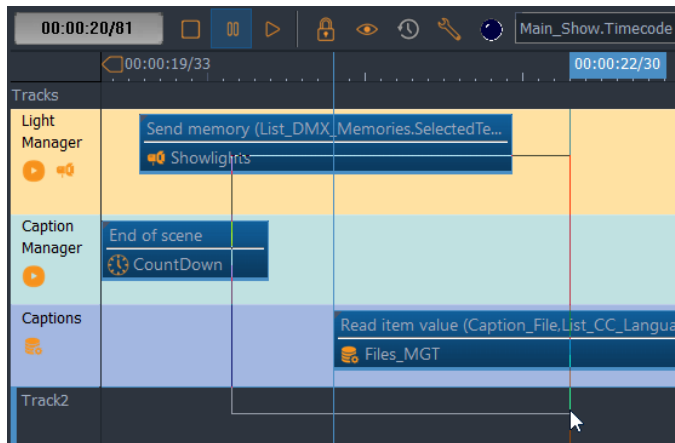
Grouping Cues

Cues may be grouped while in the time based task's Track View Mode to ease cue handling. Once a group of cues is created, clicking on any cue of the group selects all members of the group. This is particularly handy to make sure that the time relationship between cues of a group is not modified by mistake while dragging cues.

To group cues:

- drag out a marquee over the cues you want to group:

Show Programming



or:

- select the cues to be grouped and either choose the Edit/Group menu item or type *Ctrl+G*.

To ungroup a group of cues:

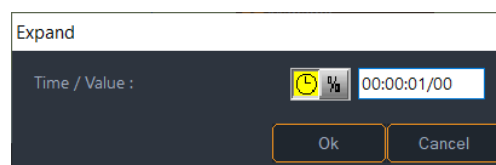
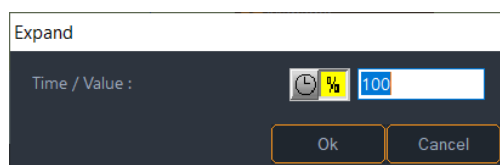
- click on one cue of the group to select the whole group and either choose the Edit/Ungroup menu item or type *Ctrl+U*.

Sizing Cues and Cue size options

The width of a cue on the timeline has no bearing on duration of the event it triggers. Cue size is flexible to allow you to see the full content of the text in the cue flag. Only the left edge is of importance (that represents the time position of the cue on the timeline). You can change the width of a cue to its default size or to the size required for all the text it contains to be fully displayed.

Expanding and Compressing Cues

Expand the time between the selected cues by a percentage or by time.

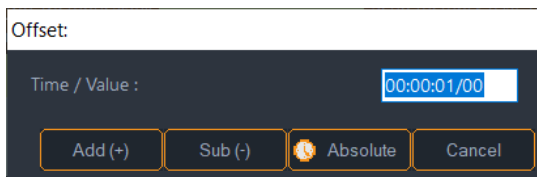


Add X percent of time to the selection, or add X second(s) to the selection.

Compress the time between the selected cues by a percentage or by time.

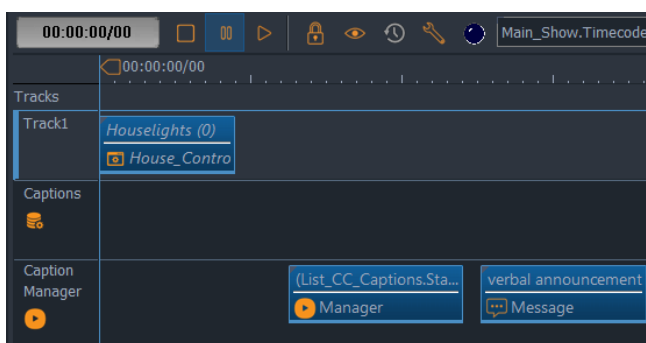
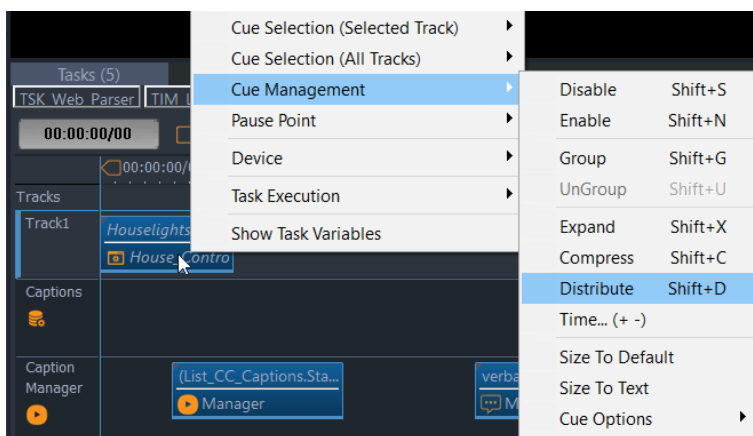
Cue time... (+/- keyboard shortcuts)

Enables you to move, relatively or absolutely, the time position of several cues by increasing or decreasing the time.



Distributing Cues

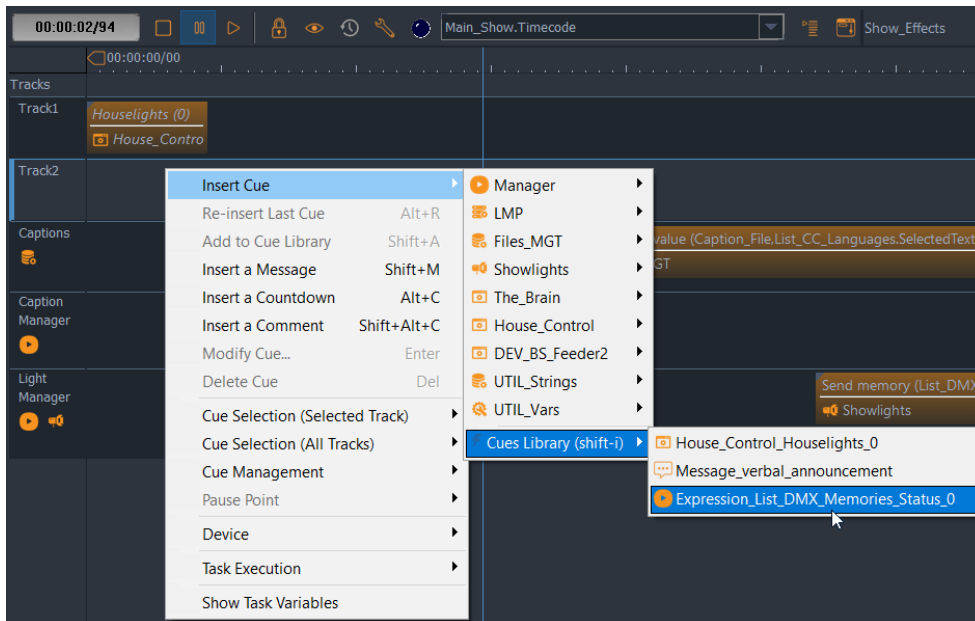
Multiple cues can be selected by holding Ctrl and clicking the cues. Distribute selected cues evenly on the timeline between the time position of the first selected cue and the time position of the last selected cue. Right-click a selected item for cue management:



5.2.7 The Cue Library

The Cue Library is a convenient way to store cues which are often used. Cues can be easily added to the library and retrieved from the library by typing keyboard shortcuts. See also [Cue Library](#). Cues in the Cue Library are stored with command-specific Properties.

Show Programming

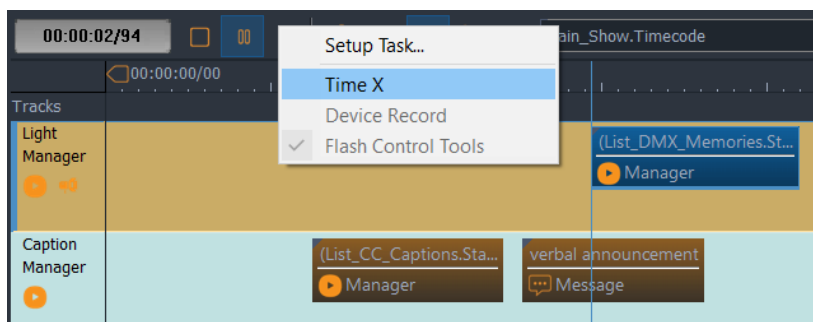


5.2.8 Time X

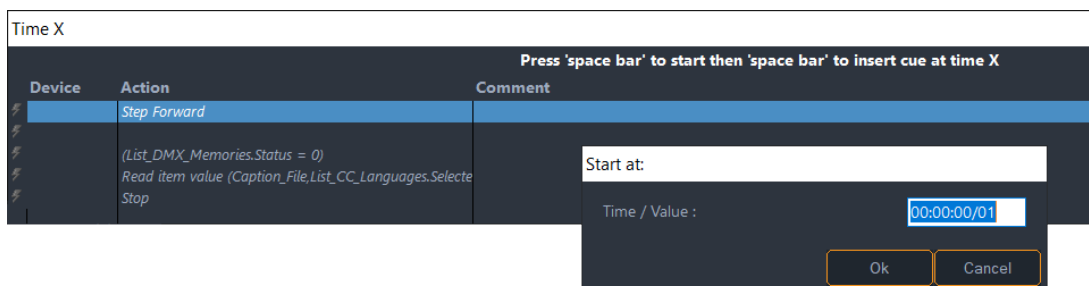
'Time X' enables selected cues to be placed on the timeline by 'tapping in' the timing while the timeline plays, i.e.: when listening to a musical rhythm or whilst watching a video.

This function works as follows:

1. Select the required cues either by dragging a marquee with the left mouse button pressed down (start in an empty area of a track), or by selecting them one by one with the mouse pointer while holding the Ctrl key.
2. Click on the setup menu button and select Time X.



3. A dialog box will ask for the start time of the distribution. When the field has been filled in, press the Spacebar to start the timeline running.



Show Programming

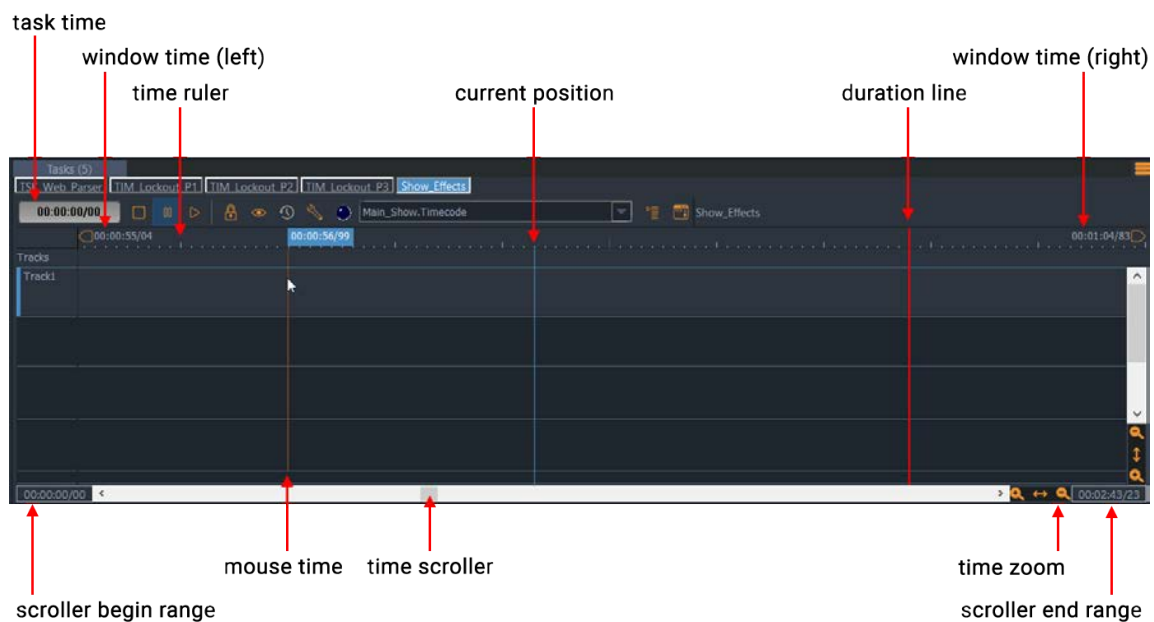
4. With each press of the Spacebar, cues are placed in the order they were in on the timeline before their selection.
5. When all cues are in place, stop the timeline.



Due to the Positrack engine, a Start time of 00:00:00/00 is not allowed. Use 00:00:00/01 to begin at the beginning of the timeline.

5.2.9 Timeline Management

There are several indicators, controls and menus that allow you to modify the current timeline's time, representation, and what part of the timeline is displayed. Below is a quick summary of their uses.



Indicators

- Task Time: displays the current time of the timeline. Clicking on this control allows you to type in a new value to locate a stopped or paused timeline to a new location (Keyboard Shortcut = G).
- Current time position: this thin blue vertical bar shows the current position on the timeline.
- Mouse time: the thin line and associated flagged time display shows the time at the mouse position.
- Task End Time: this thin red line shows the task's end time.
- Left window time: time of the leftmost displayed part of the timeline.
- Right window time: time of the rightmost displayed part of the timeline.

Controls

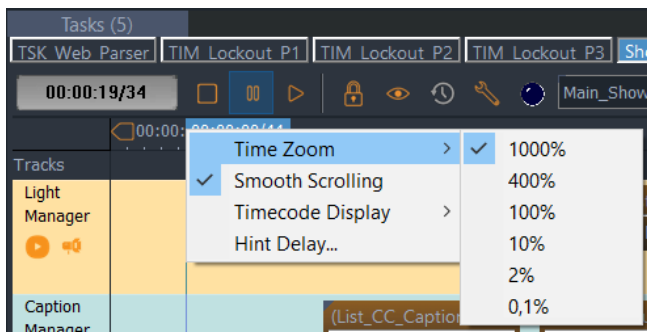
- Time Scroller: scrolls the visible part of the timeline (but does not change the current timeline time).
- Time Zoom: sets the zoom level of the view.
- Scroller Begin Range: Sets/Displays the Begin Range of the Scroller Bar.
- Scroller End Range: Sets/Displays the End Range of the Scroller Bar.

Time Ruler Contextual Menu

Right-clicking on the Time Ruler displays the Time Ruler contextual menu which allows you to set the time display format, zoom levels and other settings.

Time Zoom

Allows you to set the Zoom level of the timeline from 0.1% to 1000%, and to adapt the span of the timeline to your needs.

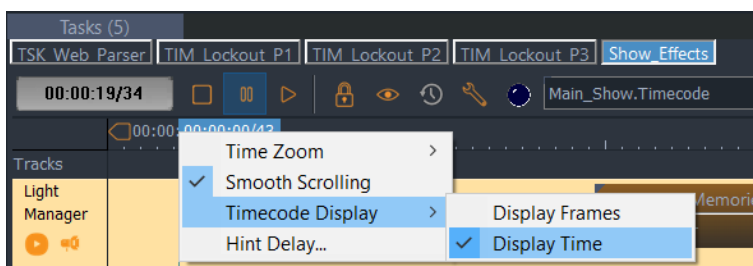


Smooth Scrolling

When Smooth Scrolling is active, the current time position is visualized at a fixed point as the timeline scrolls by. When Smooth Scrolling is off, the current timeline position moves along a static timeline, and the timeline refreshes when the cursor moves off the display. Smooth Scrolling requires more graphical resources and can be turned off to improve graphical responsiveness.

Timecode Display

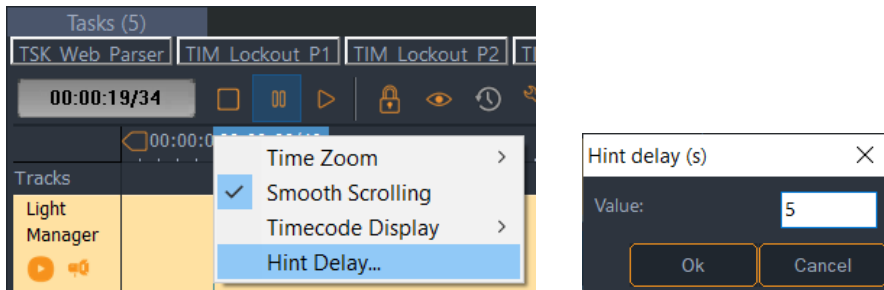
You can change the time display format between Time and Frame views.




Hint Delay

Cues can be resized to the point that the description is no longer visible.

The hint delay allows you to set the time, in seconds, before cue information is displayed when the mouse hovers over a cue (default is 1 second).



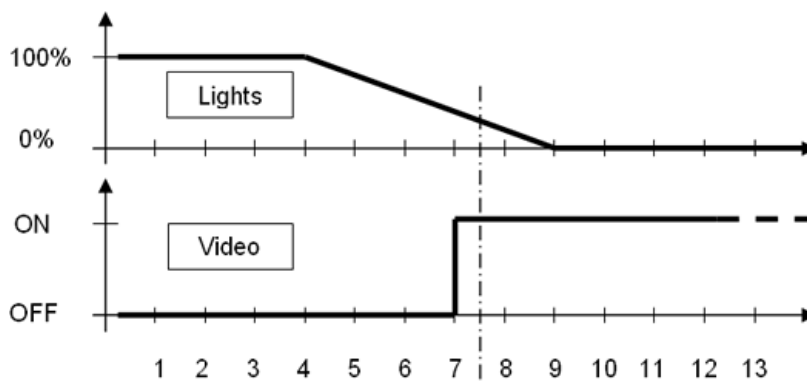
5.2.10 Positrack

Position Tracking (or 'Positrack')  is a mechanism that enables a timeline to recalculate the state of all objects controlled by a timeline ought to be at any moment in the timeline.

Positrack Mechanism

When Positrack mode is enabled, locate the timeline to the desired time (for example, by clicking on the Time Ruler) and all devices compatible with Positrack are set to the correct state.


This feature is especially useful for long shows where it isn't practical or convenient to restart the show from the beginning to check something at the end of the show.

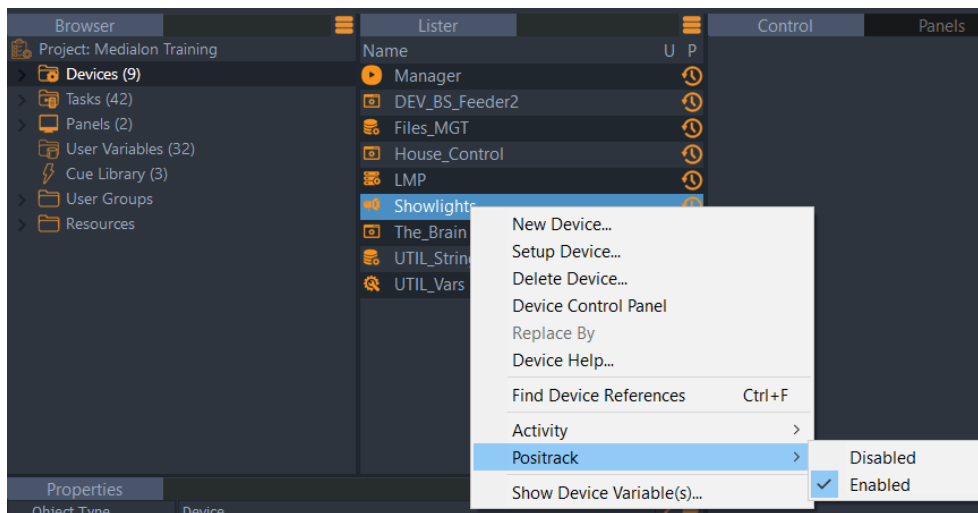


In the above illustration, 'Positrack' is engaged, following a return to time T, enabling the lighting to be reset to 30% and the video player to be repositioned at 0.5 second from the start of its sequence. Medialon Manager recalculates the states at a given time, but the speed at which the return to these states is made, depends on the type of machine being controlled.

Show Programming

This 'Positrack' mechanism enables you to reduce programming time of a sequence of events when it is necessary to run this sequence several times to adjust each event (here the position of the object) whilst enabling you to restart the sequence at any time position.

- Positrack can be enabled or disabled for the whole timeline with the 'Positrack' button  located on the task toolbar.
- Positrack for a device can be enabled or disabled by right-clicking on a device wherever shown:



Stop Task Behavior

If a task is in pause and Positrack is active, when you stop the task, devices could have different behavior. Choice of this behavior can be made in the Medialon Manager Preferences. See also [Preferences](#).

Default behavior

When you stop a task in pause, all devices used in this task, stay in their current status defined by the Positrack. A player will stay in pause, DMX channels keep transition levels, etc. The timeline releases control of the devices.

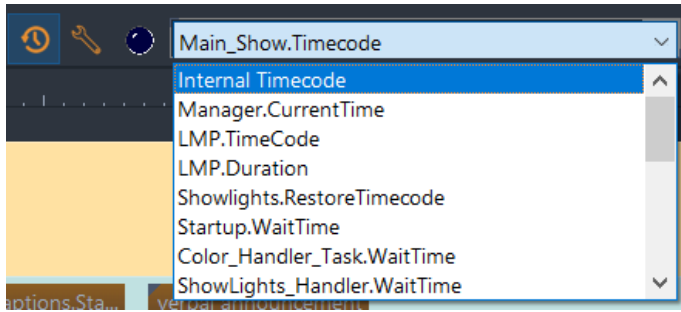
Legacy behavior

This behavior is the one used by Medialon Manager before Medialon Manager V5.1.0. or Showmaster Editor V1.5.0.

When you stop a task that is Paused, all devices used in this task are released of their pause status. All devices resume the last command sent before the task was paused. Players restart playback, DMX cues finish their fade, etc.

5.2.11 Timecode

Time based tasks use timecode as a clock reference to execute cues. The clock reference may be either internal or external. The choice is made via a popup menu at the top of the Task Editor.



Internal Timecode

When using internal timecode, the reference time of the timeline is controlled by Medialon Manager based on the internal computer or Showmaster clock. If the timeline is running, the reference time increases. If the timeline is paused, the reference time is frozen, etc. This is the default mode of the timeline.

External Timecode

You can have the timeline synchronize to any Time variable available in the current project. The Timecode pull down menu displays all time variable sources available.

Several settings on the Task Setup dialog box define the behavior of the timeline regarding timecode drop-out or jumps.

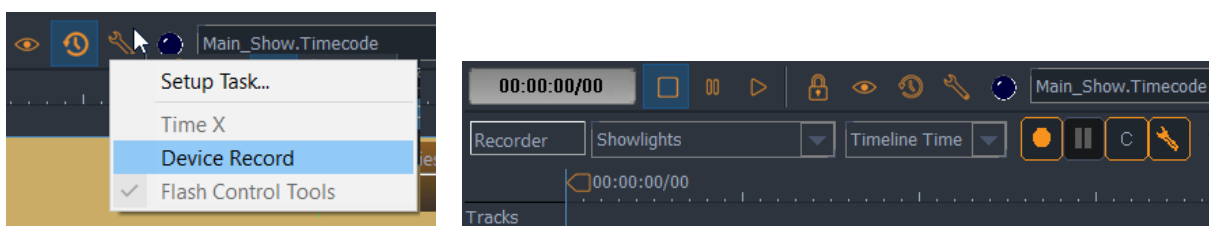
5.2.12 Recording Cues

The device recording function of Medialon Manager V7 allows you to record information from certain devices directly onto a timeline.

The following MxMs are compatible with Device Recording:

- MxM Medialon_Basic_MIDI_In_Out
- MxM Medialon_MIDI_Show_Control
- MxM Medialon_Low_Level_Communicator
- MxM Medialon_DM512
- MxM Medialon Variables Management

The Device Recording process is activated using the setup button in the task's toolbar if the selected device includes this feature.



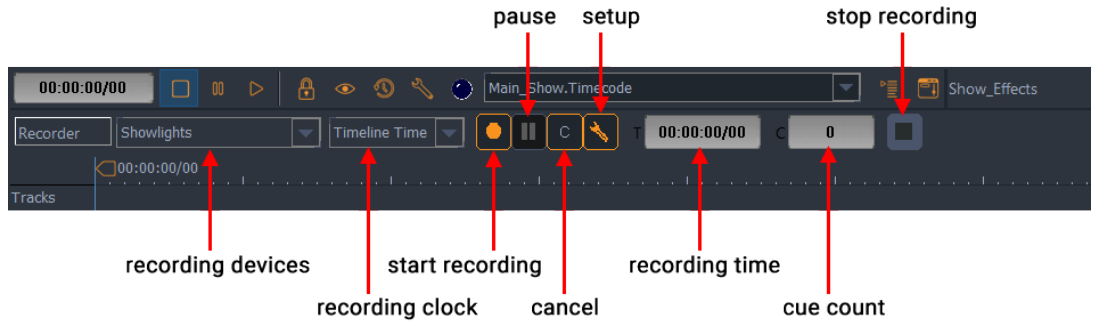


Note that if the device does not support cue recording capability, the Device Record button is not accessible.

Cues are recorded by the device and inserted in the task at the end of the recording session. Cue contents depend on the device type.

Device Recorder Tool Bar

The Device Recorder has its own toolbar.



Recording Devices

List of Devices used by the recorder. If multiple devices are selected, settings for each device's recording can be set by choosing the device from this dropdown selector.

Recording Clock

- Medialon Manager V7 can use the current timeline's timecode as the clock reference during the recording. The recording is captured when the time based task is running.
- Medialon Manager can use the Recorder's Internal Time. In this case, recording is captured even if the timeline is stopped.

Setup Recording

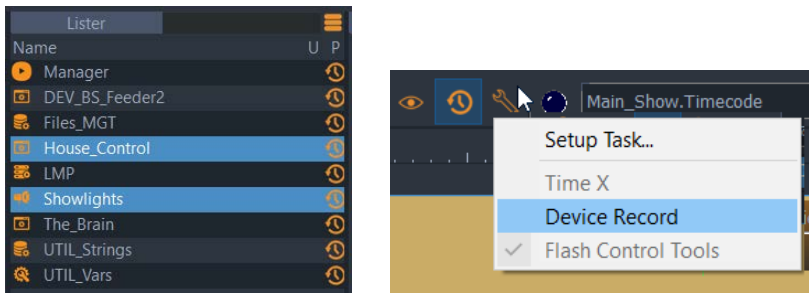
This menu allows selecting the type of recorded cues you want. Choices are different depending on the selected device's type.

DMX Device Recording

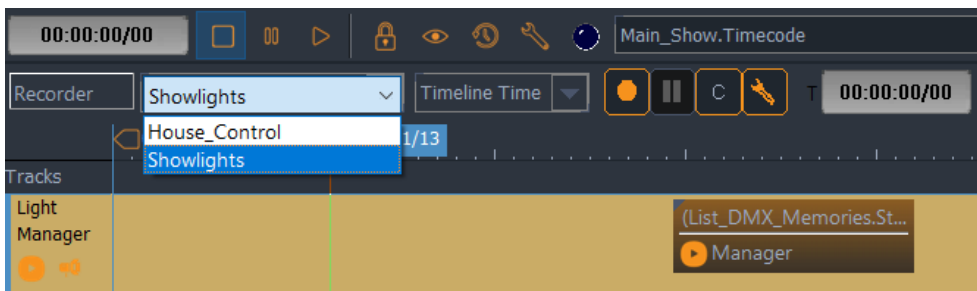
To start a DMX recording, follow steps below:

Show Programming

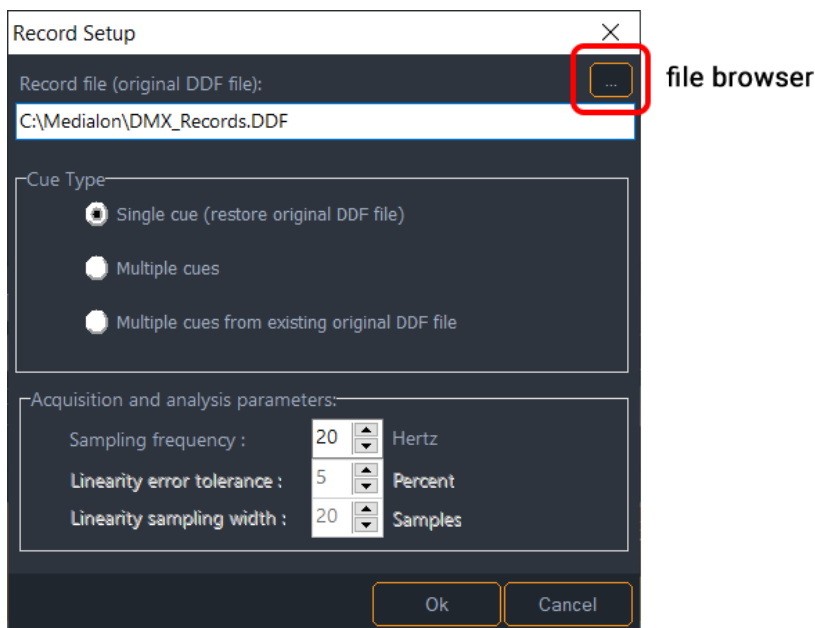
1. Select DMX device to be recorded. Multi selection is allowed.



2. Open Recorder Toolbar using Tools/Device Record button in the Timeline Editor.
3. The recording needs to be configured for each selected device, click on Record Setup button.



4. Record Setup window will appear, you may choose to record the Data as one single cue, or as multiple cues:

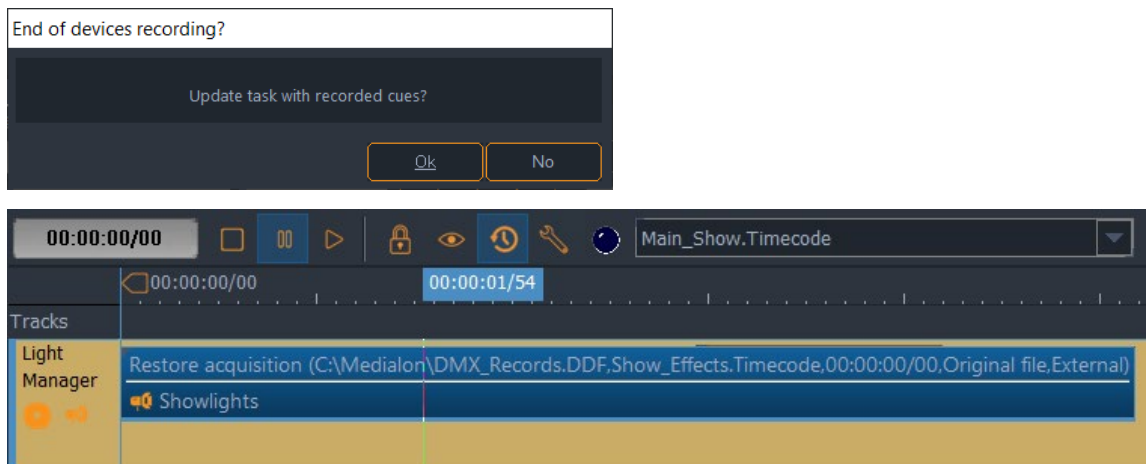


If you select 'Single Cue', remember to define the record file path and name. Be sure to use the .DDF extension on the file name.


5. Select the start position on the task (timeline) where recorded cues will be inserted.
6. Press the Start Recording Button. *Ctrl+Click* to start the recording of all selected devices.

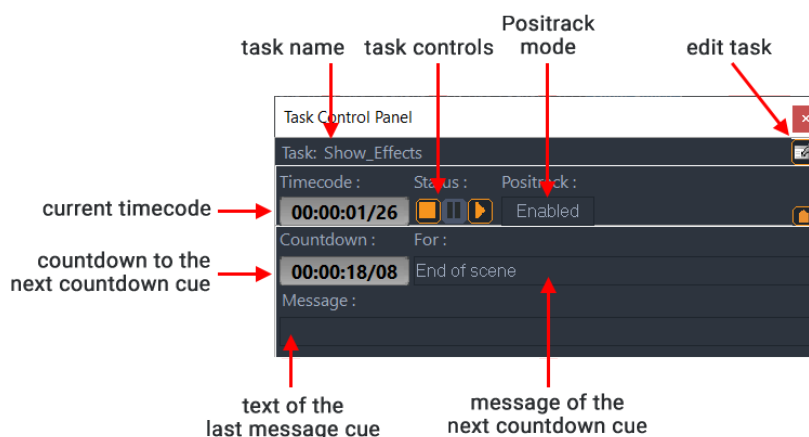
Show Programming

7. Start the timeline to activate the recording process.
8. At the end of the recording, press Stop Recording button. Medialon Manager V7 updates the timeline with the recorded cues.



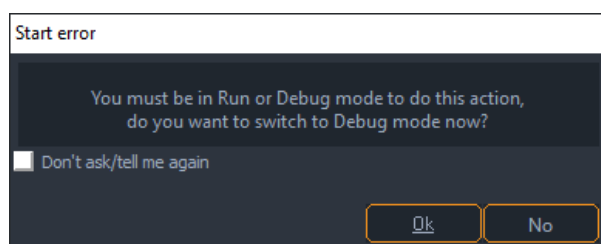
5.2.13 Controlling Tasks

The Task Control Panel  shows the main runtime information and commands of a time based task. It is displayed by clicking on the 'Control Panel' button in the Task Editor window. When the Control Panel window is resized, inner controls are resized automatically to ease far away reading.



With the Task Control Panel, you can start, stop, and pause your timeline. Medialon Manager V7 needs to be in Debug, User, or Running mode to start a task.

Switch Medialon Manager V7 to Debug Mode by pressing the *F2* key.



Show Programming



A warning dialog box asks you to switch to Debug mode if Medialon Manager is in Stop Mode and you try to start a task.

Use the Task Control buttons

- Click on the task 'Play' button to start the show.
- Click on the task 'Pause' button to pause the show.
- Click on the task 'Stop' button to stop the show.



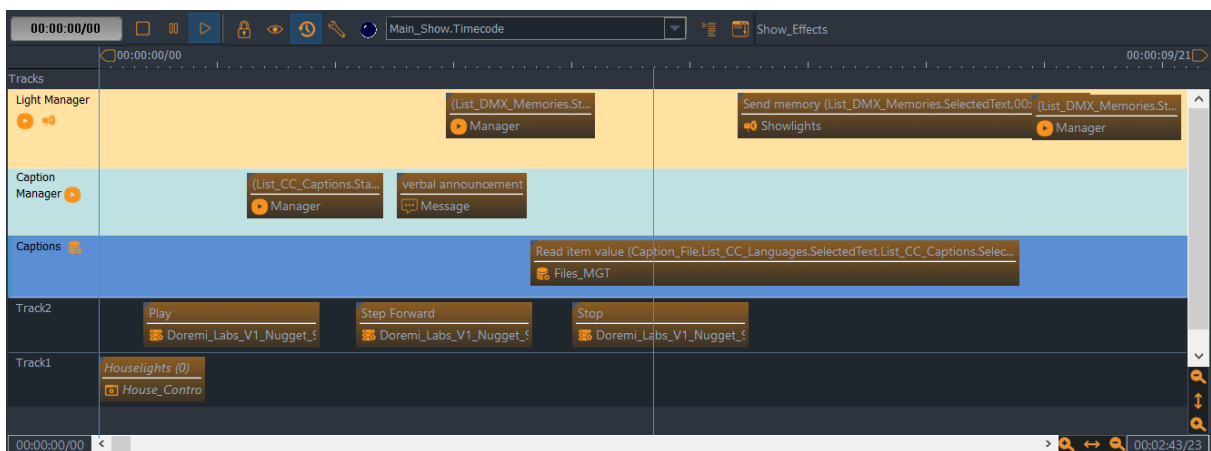
Note that the timeline stops automatically after the last cue and will not run if there are no cues are inserted unless the task duration has been increased in the Task Setup.

5.2.14 Views

The task editor offers two views for time based tasks. Clicking on View button toggles between timeline view and List view.

Timeline View

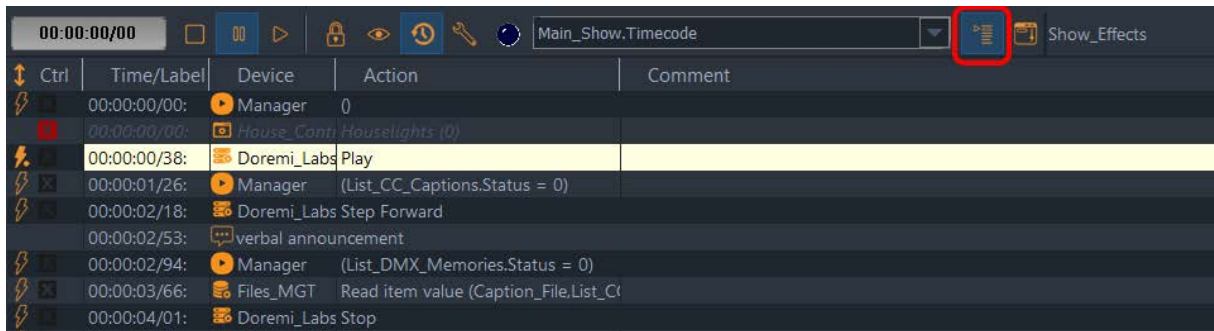
In the timeline view, cues are organized in tracks and time position. A time ruler allows you to move cues around in the timeline.



When the timeline runs, tracks scroll from right to left; the current time position stays in the center of the window.

List View

The list view shows the time based task as a cue list.



The cue list scrolls down when the task is executed. In this view, ‘Flash Control Buttons’ are present.

5.2.15 Flash Controls


The ‘Flash Controls’ provide a mechanism for controlling the cues in a task regardless of the current position or status. Flash controls are available even if the task is not running.

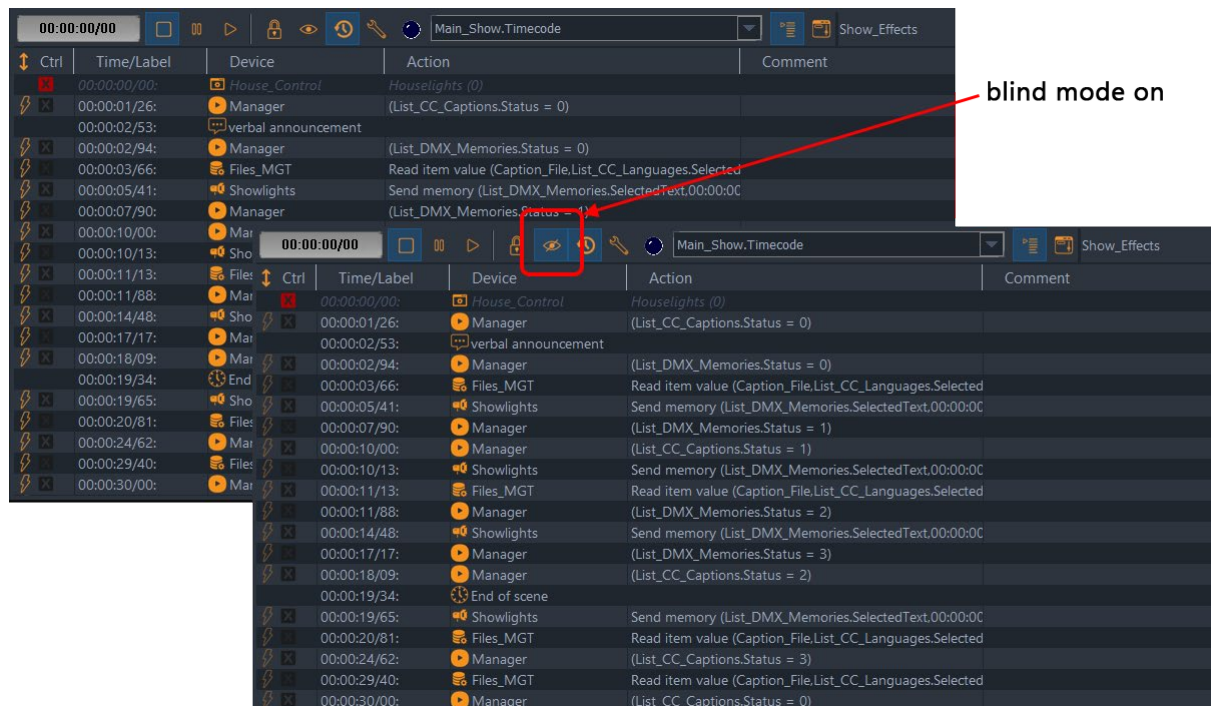



The Flash Button on the left of each Command Cue lets you execute that cue. The cross button allows you to disable (red) or enable (grey) the cue.

‘Message’ and ‘CountDown’ Cues do not have Flash Controls.


5.2.16 Blind Mode

Blind mode  (on) allows you to run and edit your task without executing any of the cues. This is useful when you need to edit a timeline without having Positrack control the devices (for example, when another show is running).



When blind mode is On , cues cannot be executed even with Flash commands. The blind mode affects the selected task; all other tasks in your project continue to execute as normal, devices perform all commands received by the other tasks.

5.2.17 Lock Task

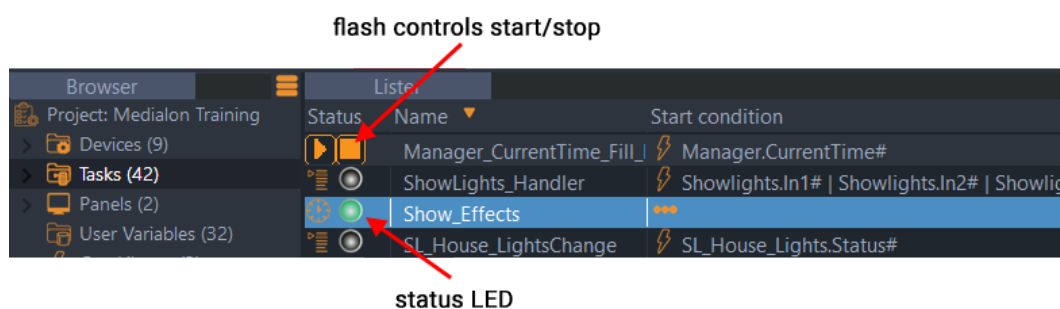
Lock Task  prohibits the task from being launched automatically when the start condition becomes true or when a Start Task command is sent. The task may still be fired by using the play buttons in the Task Control Panel and Task Editor Tab, or with the Flash Controls.

5.2.18 Pause Points

Pause Points allow you to step through your tasks for debugging complicated tasks. This function is not available in Run Mode. See also [Advanced Tools](#).

5.2.19 Tasks in a Project

All tasks created in a project are listed in the Browser Tab under the Tasks node.



Show Programming

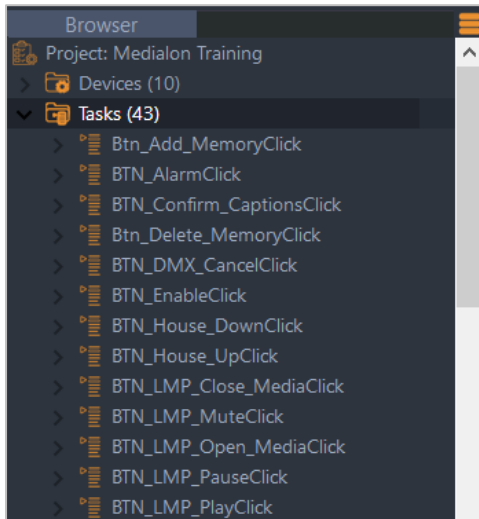
Select the Tasks node to see all tasks in the Lister.

The status of each task is displayed by a LED (Dark Blue = Stopped, Yellow = Paused, Light Blue = Running).

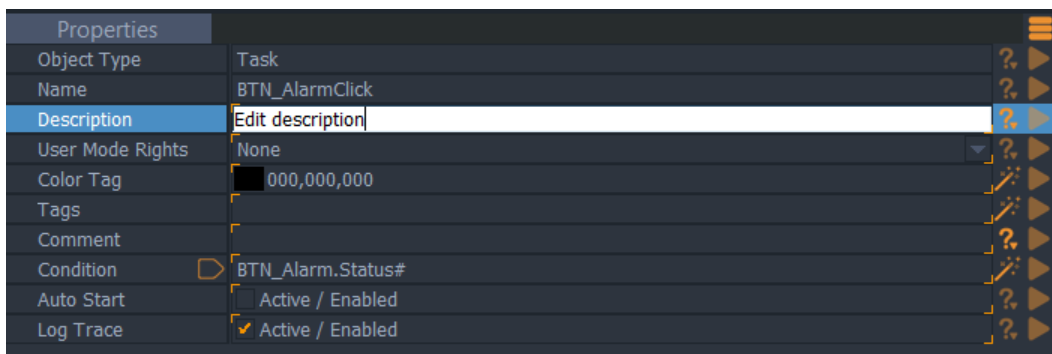
Flash controls to start and stop tasks are available in the Task Lister tab.

Task Properties

Expand the Tasks node in the Browser tab to see all tasks.



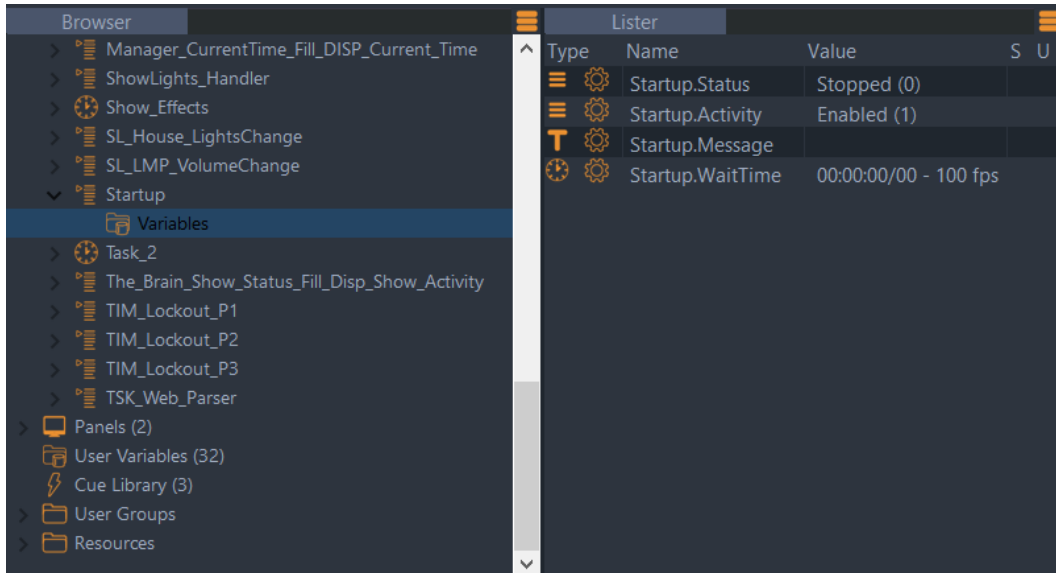
When you select a task in the Browser or in the Lister, its properties are displayed in the Object Properties tab



- Name: You can rename the task. Remember to validate your changes with the Enter key.
- Description: Description of this task
- User Mode Rights: Defines if this task is visible or editable in User Mode
- Comment: You can add a comment about this task
- Condition: Condition to automatically start the task. This is an expression; when it evaluates as true (any value other than 0), the task starts if it is not already running.

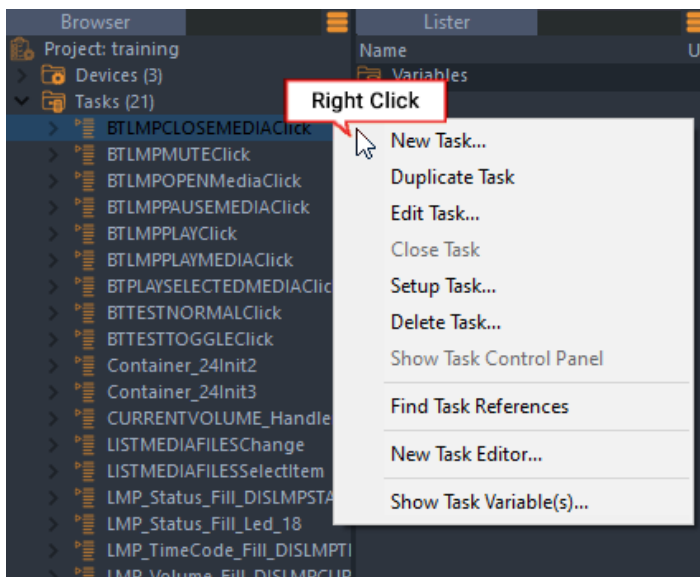
Task Variables

Select a task and expand it in the Browser tab to display the list of variables or the tracks in the Lister tab.



Task Management

Once you have created your task, you may access the task management menu by right-clicking on the desired task in the Browser tab or in the Lister tab. The contextual menu appears, giving you access to the management options.



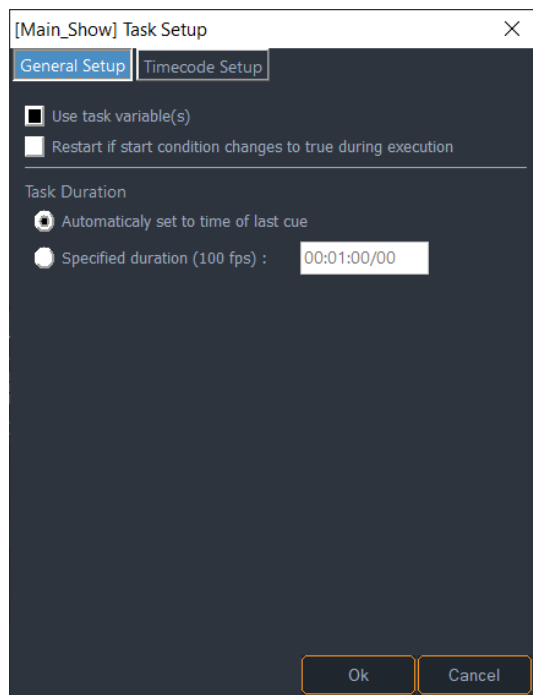
- *New Task*: Creates a new task
- *Duplicate Task*: Duplicates the selected task and names the duplicate with a numerical incremental value

Show Programming

- *Edit Task*: Opens the task editor in the Task tab or switches to the selected task's editor if it is already open
- *Setup Task*: Opens the selected task's setup window
- *Delete Task*: Deletes the selected task from the project after prompting for confirmation
- *Show Task Control Panel*: Only available for time based tasks, this command opens the selected task's Control Panel inside the Control Tab
- *New Task Editor*: Opens a new Task tab. With the new tab open, selecting Edit Task opens the Task Editor of the selected task in the new tab.
- *Show Task Variable(s)*: Displays variables of the selected task in the Lister

5.2.20 Task Setup

Clicking on the Task Setup button or right-clicking on the task in the browser and selecting 'Setup Task...' opens a Task Setup dialog for the currently selected task.



General Setup

Use Task Variables

If this option is checked, task variables are created.

Restart if start condition changes to true during execution

If this option is checked, the task restarts when it reaches the end if its Start Condition changes to true while the task is running. This is useful to guarantee that a variable change which causes the condition to become true is correctly taken into account inside the task.

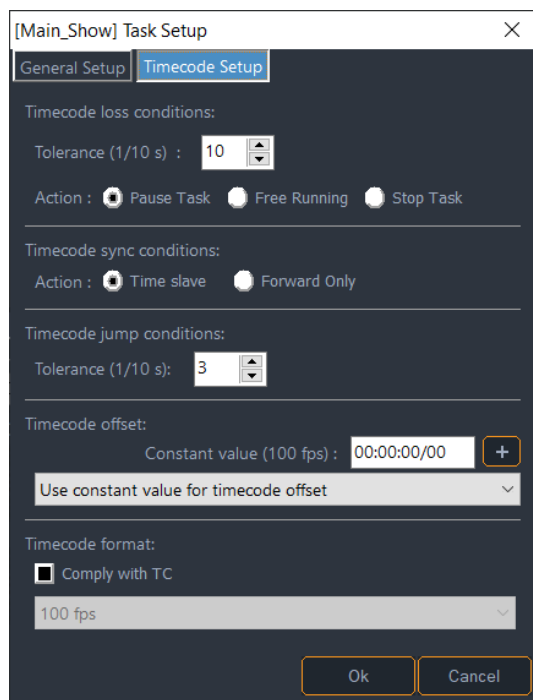
Show Programming

Task Duration

Specifies the duration of the task.

Timecode Setup

An additional tab is displayed on the Task Setup dialog for time based tasks. This tab allows you to configure the Positrack behavior of the task depending on its timecode reference.



Timecode Loss Conditions

Defines the behavior of the task when reference Timecode is lost (Timecode stops). The tolerance is the time (in tenths of a second) between the expected position of the Timecode and the timeline's actual position. A value of 10 means that the task will consider that the Timecode is lost if the gap between the reference Timecode and the expected Timecode is greater than 1 second.

The Action parameter defines what the task will do if the Timecode is lost:

- *Pause*: the task pauses
- *Continue*: the task ignores the incoming timecode and continues on its internal clock (Free Running)
- *Stop*: the task stops

Timecode Sync Conditions

- *Chase Timecode*: allows continuous chasing as timecode jumps around (follows the incoming Timecode)

Show Programming

- *Forward Only*: enables you to chase the Timecode only if the source Timecode is greater than or equal to the timeline's current time position. For example, if you don't want the Positracked Devices to reposition themselves each time the reference Timecode goes back, then choose this option, in which case the timeline will wait for the reference Timecode to reach the value of its position before switching to 'Run'.

Timecode Jump Conditions

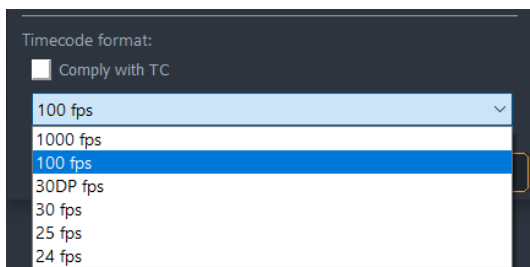
Defines the time tolerance (1/10 second) before considering that the reference Timecode has skipped forward.

Timecode Offset

Defines an offset to add to the external Timecode reference incoming value. It can be a positive or negative constant value or any time variable. Using a time variable allows the timeline to be synchronized to an external timecode source like a GPS synchronized timecode generator, but have the actual show time be dependent on the value of the Time variable used as the offset.

Timecode format

Allows you to setup the frame rate of the task. If 'Comply with TC' is checked, the frame rate of the task will automatically adjust to match the framerate of the Timecode source.



6 Automation Programming

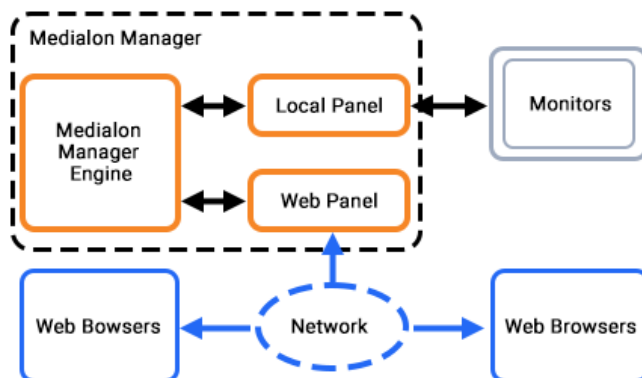
In this section you will learn how to turn a show program into an automated application and how to create Graphical User Interfaces.

6.1 Panels and UserScreens

The 'UserScreen' is the only visible part of the Medialon Manager application in Run mode, where the user will choose, make decisions, get information, play, and act on everything that is programmed in your project.

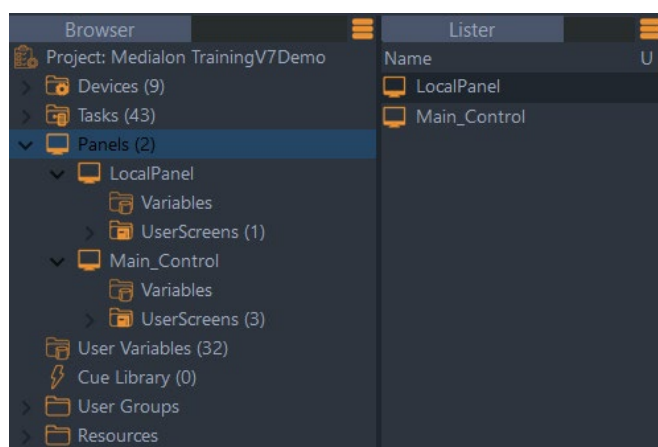
6.1.1 The Panels

With Medialon Manager V7, UserScreens can be displayed on the current monitor of Manager V7 or remotely displayed using a web browser. The display area is defined as a panel.



Local Panel

The Local Panel uses the monitors connected to the Medialon Manager machine to display the UserScreens.

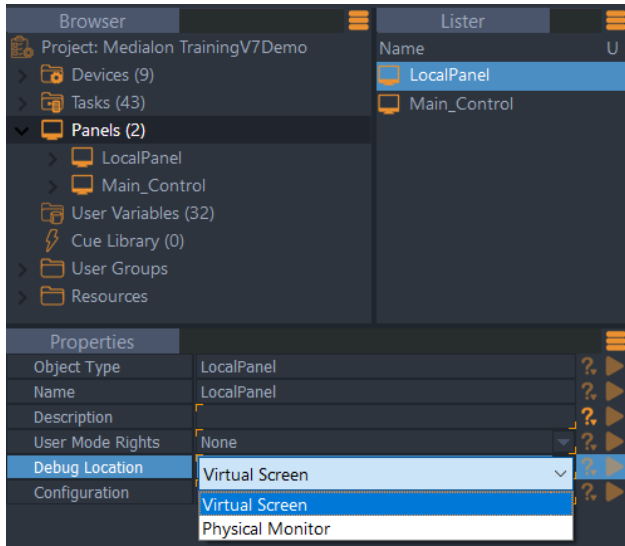


Local Panel is automatically created when a new Medialon Manager project is started.

Virtual Screen

The Virtual Screen is designed specifically for programming and debugging; Virtual Screen allows you to simulate your screen resolution while keeping all the other Areas and windows visible. Furthermore, the Virtual Screen allows you to create graphical user interfaces at a higher resolution than your current screen settings can handle, allowing you to program on a different system than the final runtime system.

There are two Debug location for the LocalPanel:



- *Virtual Screen*: UserScreens are displayed inside the virtual screen in Debug Mode.
- *Physical Monitor*: UserScreens are displayed as a separate window in Debug Mode.

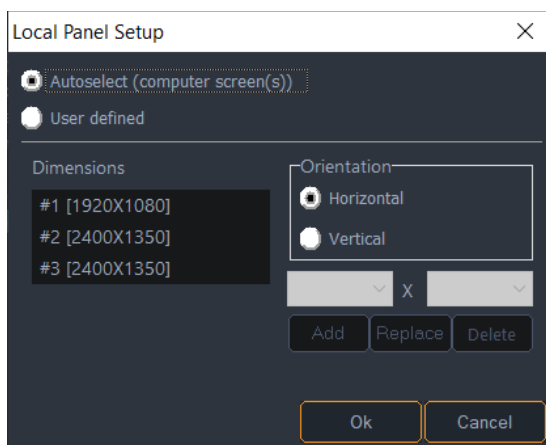


These settings do not affect the Run mode of Medialon Manager, where all UserScreens are displayed on the physical monitor.

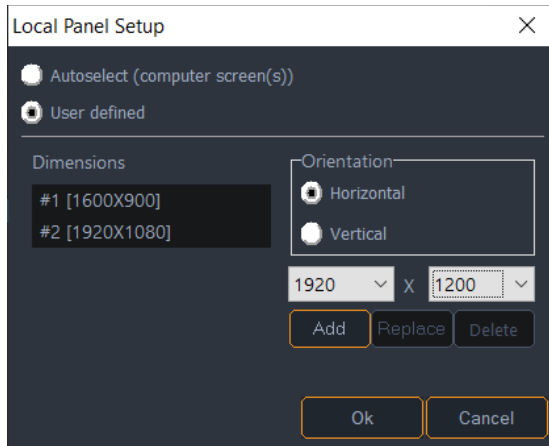
Panel Layout

Defines the Local Panel screen's resolution.

- *Autoselect*: When chosen, Local Panel takes the current screen resolution of the system.



- *User defined*: If you need to create a project that Local Panel will eventually need a different screen resolution, you can define the screen(s) resolution(s) by editing the User Defined Screen resolutions.

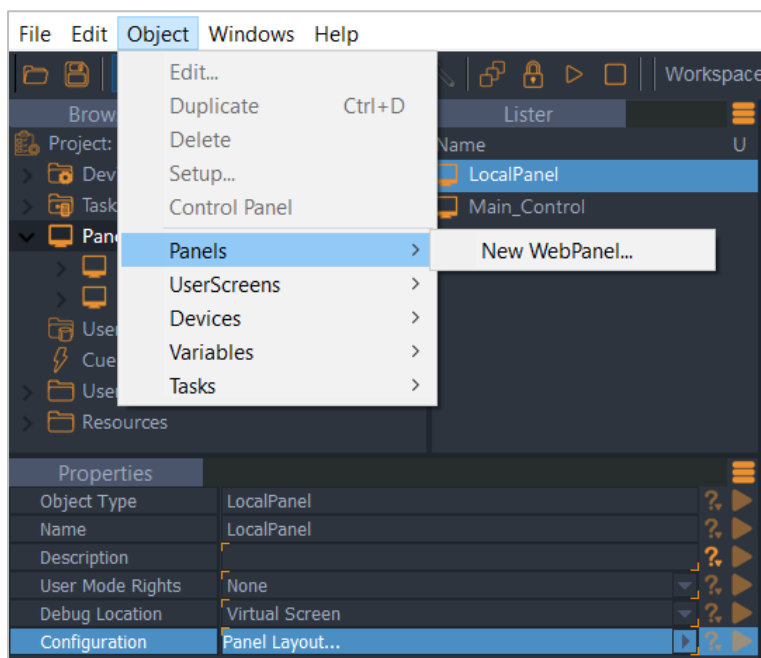


- *Orientation* defines the direction of screens alignment Select screen resolution
- *Add* a new screen
- *Replace* selected screen by new resolution
- *Delete* selected screen.

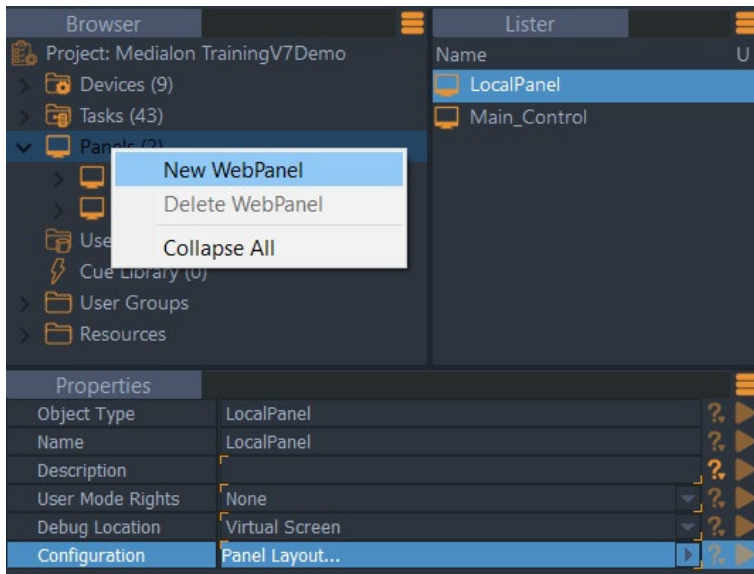


Creating a WebPanel

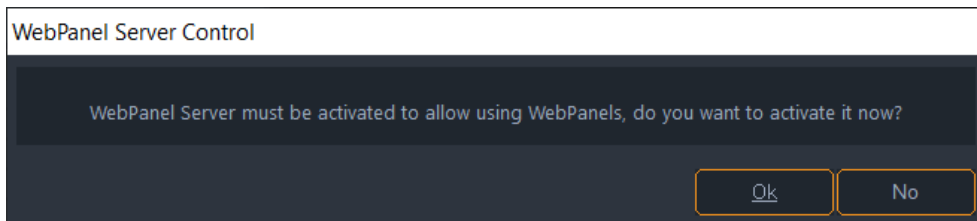
Select the 'Object/Panels/New WebPanel...' menu item.



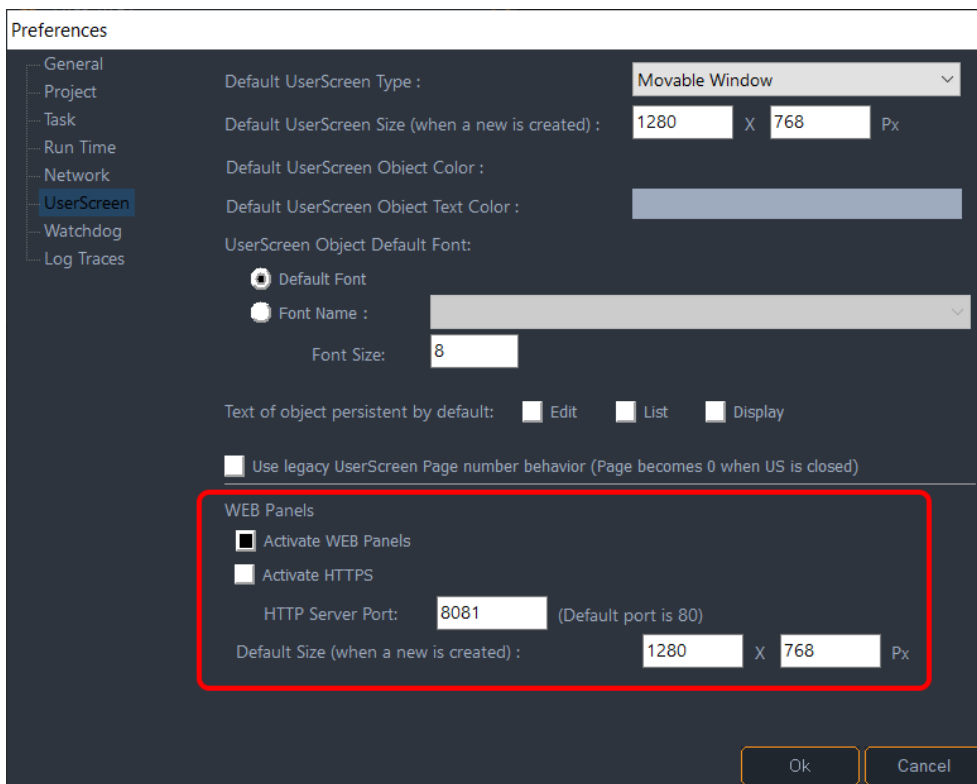
Or select the Panels item in the Browser tab and right-click to open the contextual menu.



i The WebPanel Server must be active for the WebPanels to be served to web browsers.

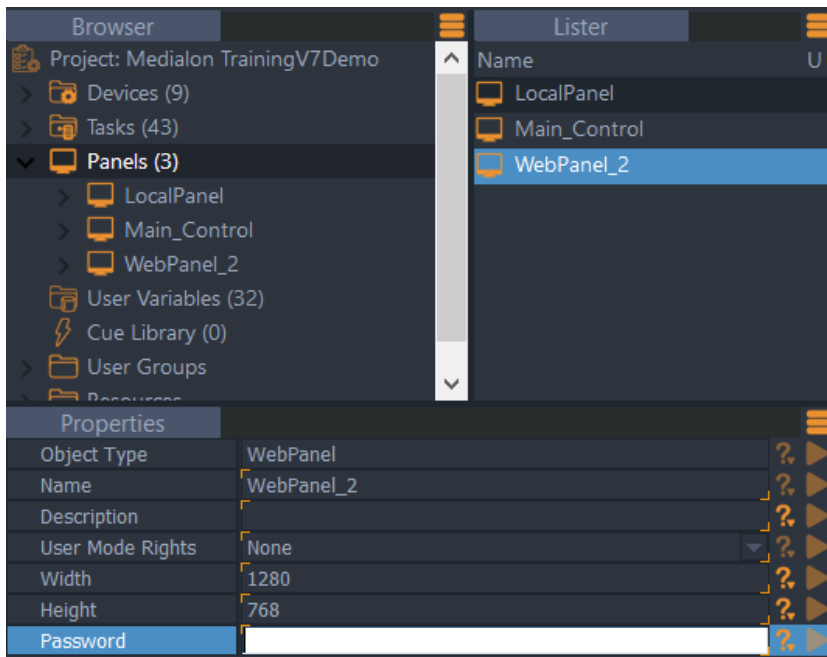


Click OK to activate. Other settings are in File > Preferences > UserScreens:



WebPanel Properties

When a WebPanel is created, you need to define the size of the pixel space to be rendered in the remote web browser.



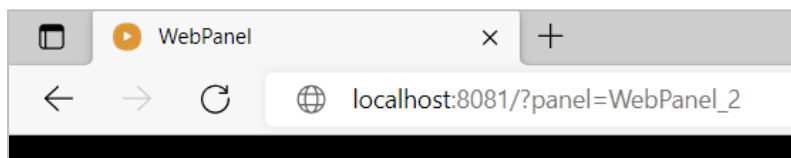
WebPanel access can be protected by a password.

Connecting a WebPanel

Open a web browser.

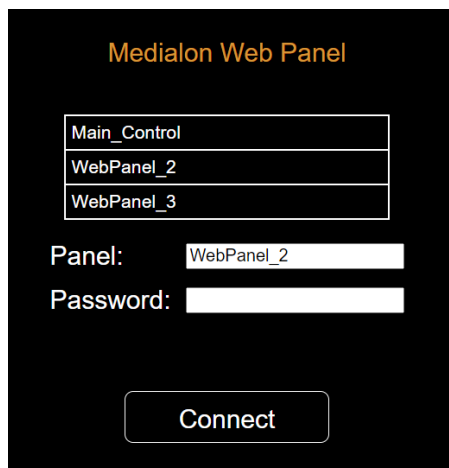
Enter `http://address[:Port][?panel=WebPanelname][&password=password]`

- **address**: IP address or Computername.
- **port**: (Optional) WebPanel Server Port used by Manager V7, default port is 80.
- **WebPanelName**: (Optional) name of the WebPanel to open.
- **password**: (Optional) WebPanel password if required.



If no panel name is specified in the address and there is only one WebPanel in the project (and the WebPanel has no password assigned), the web browser loads the WebPanel.

If more than one WebPanel exists in the project, the Medialon WebPanel login page opens:



Enter a WebPanel name or select it in the WebPanel list; type the WebPanel password (if a password has been set).

Web Browser Compatibility

WebPanel can be opened with recent versions of Google Chrome, Mozilla Firefox, Safari, Internet Explorer, Edge, Opera, Brave, Safari iOS and Android – more generally, any browser that supports HTML5 and JavaScript.

Kiosk Mode

Most of the times, you will run the WebPanel in a browser with a special mode which prevents the end user from having access to unauthorized applications and resources of the machine running the browser. This is usually done by running the browser application in 'kiosk' mode.

Kiosk mode enables a user to view web browsers in full screen. Kiosk mode is an extension or a tool or plug-in available to users to enlarge the browser window to full screen and removes features such as navigation buttons and address bar. Some examples are given below:

PanelAddress = [http://address\[:port\]?pane1=webpane1name\[&password=password\]](http://address[:port]?pane1=webpane1name[&password=password])

- *Chrome*: enter command > chrome.exe --kiosk *PanelAddress*.
- *Internet Explorer*: enter command > iexplore -k *PanelAddress*.
- *Edge (v87 or later)*: enter command > msedge.exe --kiosk *PanelAddress* --edge-kiosk-type=fullscreen
- *Firefox*: install R-Kiosk add-on and make it point to *PanelAddress*.
- *Safari iOS*: save shortcut as a home screen icon. The URL opens in full screen kiosk mode.

If you want the remote machine to automatically connect to a specific WebPanel at startup, you can set up an automatic launch of the browser targeting the WebPanel address.

Panels Variables

Panel variables provide information about the Panel. For example:

Type	Name	Value	S	U
8	WebPanel_2.KeyPress	0		
8	WebPanel_2.KeyStatus	0		
8	WebPanel_2.CtrlKeyStatus	0		
8	WebPanel_2.ShiftKeyStatus	0		
8	WebPanel_2.MouseXPos	0		
8	WebPanel_2.MouseYPos	0		
8	WebPanel_2.MouseClickLeft	0		
8	WebPanel_2.MouseClickRight	0		
T	WebPanel_2.CurrentUserScreenName			
8	WebPanel_2.UserScreenMouseX	0		
8	WebPanel_2.UserScreenMouseY	0		
8	WebPanel_2.ConnectedWebPanels	0		
T	WebPanel_2.OpenedUserScreens			

- Key pressed on the keyboard.

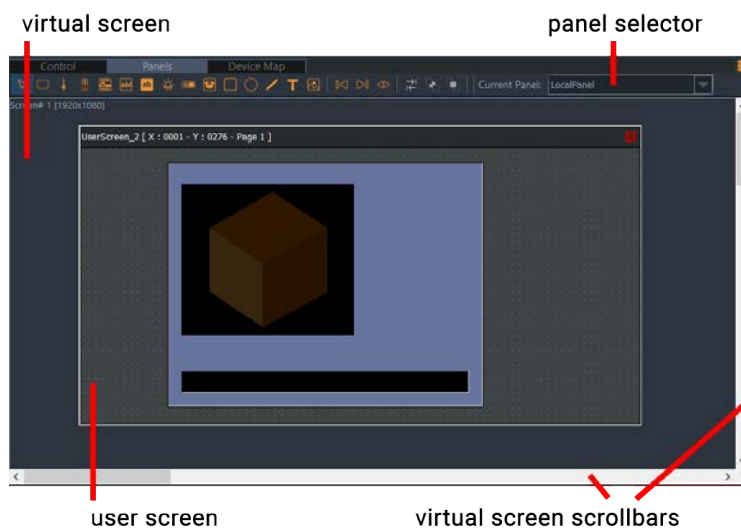
i Alt key status (only available in LocalPanel)

- Mouse position and buttons pressed
- Current UserScreen selected and mouse position in this UserScreen.
- How many WebPanels are connected.

i The number of simultaneous WebPanel connections is limited. See also [Panel/OpenCap Connection Chart](#).

The Panels Tab

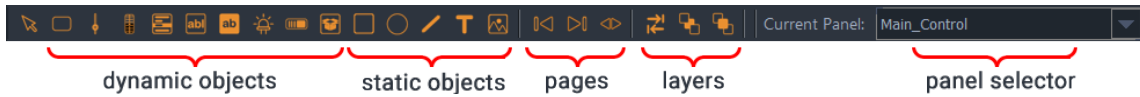
The Panels tab shows the UserScreens inside the virtual screen so you may create your interface while still having complete access to the Browser, Listers, Task Editors and Properties of your project.



The Panel Selector is how you switch between the different panels available in the project. The UserScreens are represented by windows inside the Virtual Screen.

Panels Toolbar

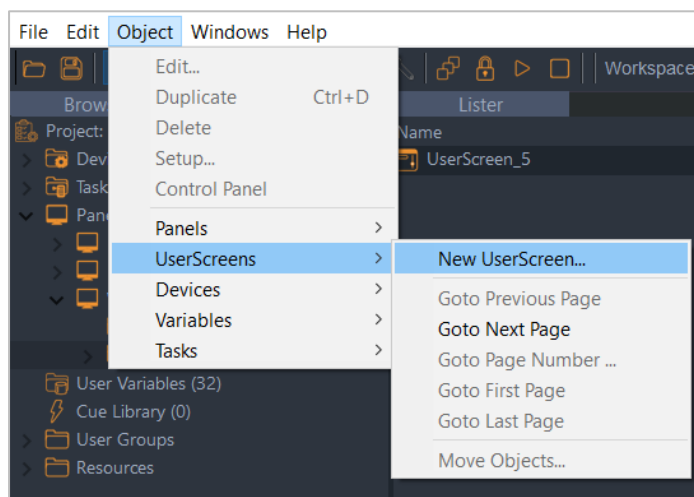
The upper part of the Panels Editor contains the UserScreens toolbar.



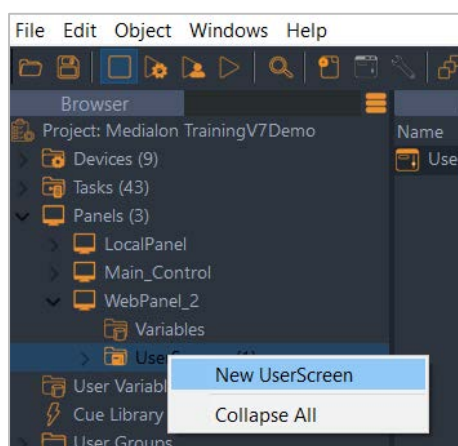
- Dynamic and static graphical objects that can be drawn on a UserScreen.
- UserScreen page selector.
- Graphical objects layer and alignment control.
- Panel Selector.

6.1.2 Creating a UserScreen

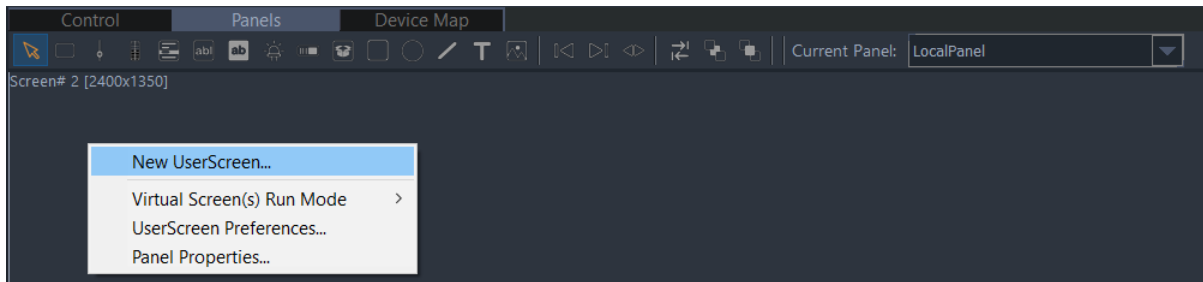
Select the 'Object/UserScreens/New UserScreen...' menu item.



Or select UserScreens in the Browser tab and right-click to open the contextual menu.

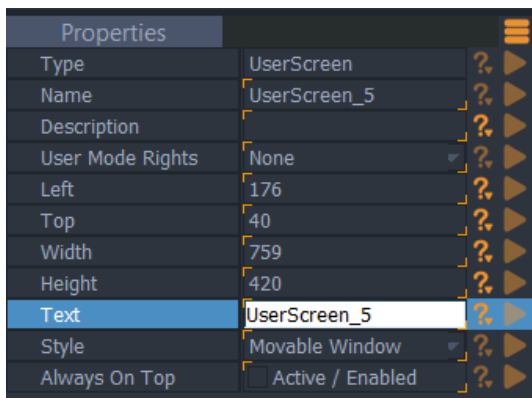


Or right-click inside the Panels tab. Medialon Manager automatically opens the new UserScreen in the Panels tab in the context of the currently selected (or most recently selected) LocalPanel or WebPanel.



UserScreen Properties

The Properties tab displays the UserScreen's settings.



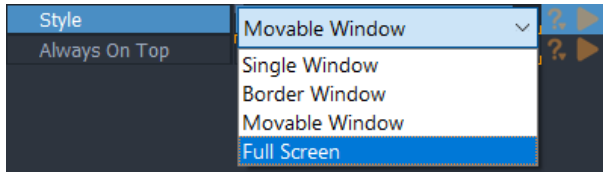
- *Object Name*: Name your UserScreen; meaningful names can help you remember what objects you are programming.
- *Description*: Enter extra information.
- *User Mode Rights*: Defines if this UserScreen Object is visible in User Mode
- Position of upper left corner (*Left / Top*): Use the mouse to move the UserScreen, or you can enter new values in the properties fields.
- The size of the UserScreen is defined by its *Width* and *Height*. The mouse can be used to resize the UserScreen, or you can enter values in the fields.
- You can customize the *Text* displayed in the title bar of the UserScreen.
- When the '*Always on Top*' option is active, the UserScreen will always be in front of all other UserScreens if they overlap.



When UserScreens are opened in Virtual Screen (Debug or User Mode), 'Always on Top' option is not active.

UserScreen Style

A UserScreen can be displayed with 4 UserScreen styles, the choice affects the way the UserScreen appears in Run Mode.



- *Single Window*: The UserScreen is painted in a fixed position window without a border. The user cannot move it.
- *Border Window*: This style adds a thin border to the Single Window, and still cannot be moved.
- *Movable Window*: The UserScreen has a title bar and borders. The user can move the window on the screen. It is the default style when you create a new UserScreen.
- *Full Screen*: With this style, the UserScreen fills the physical screen. This option is useful to hide and protect the computer's desktop.

UserScreen Pages

A UserScreen can be composed of several pages (up to 999). When you create a new UserScreen, the first page is automatically generated.

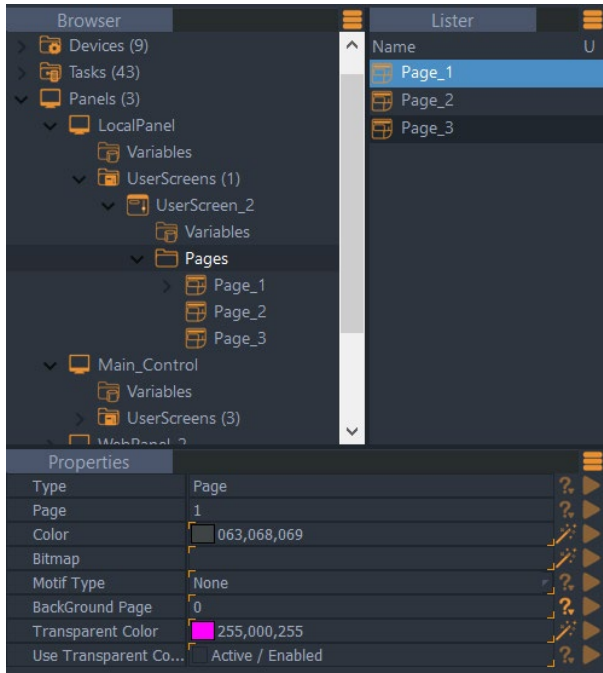


The UserScreen Toolbar has buttons to navigate among the pages of a UserScreen.

Use the 'Next Page' button in the Toolbar to add new pages. Once pages are created, they cannot be deleted unless the entire UserScreen is deleted. There is no harm in having more UserScreen Pages than you need.

Page Properties

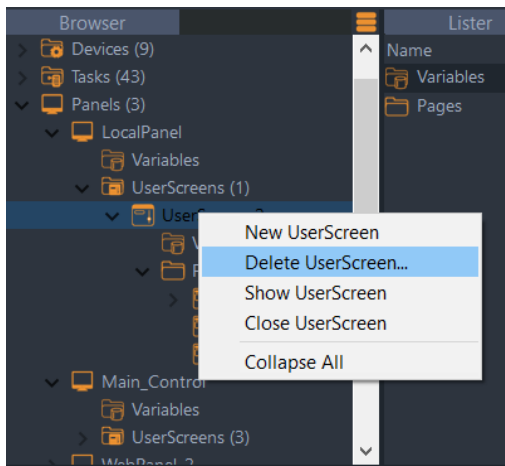
Page properties are displayed in the Properties tab by clicking in the UserScreen's page or by selecting the Page in the Browser or Lister. A page can have another page as its background. If a background page is set, all objects on the background page are visible on the current page and functional in Debug or Run modes.



- **Color:** Defines the background color of the page; use the magic wand on the right side to open the Color Selector box.
- **Bitmap:** To select a background bitmap, use the magic wand to open the Image Selector (.bmp, .jpg or .png – including transparency – can be used).
- **Motif Type:** Depending of the background bitmap you select, you can define the type of motif used (Stretch / Pattern).
- **Background Page:** the same Objects can be used on several pages. Instead of duplicating the objects, you can use a page as background. All objects present in the background page are displayed and active in the current page.

UserScreen Management

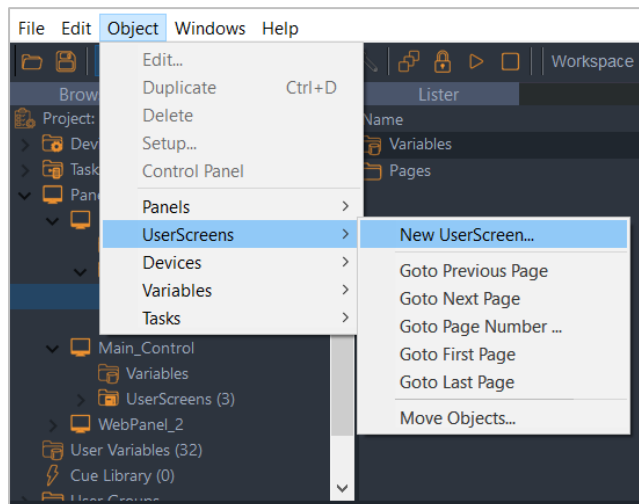
To manage UserScreens once they have been created, you can access the Management menu by right-clicking on the UserScreen inside the Browser tab or inside the Lister tab.



From the Browser or lister tabs:

- *New UserScreen*: Creates a new UserScreen
- *Delete UserScreen*: Deletes the selected UserScreen from the Project
- *Show UserScreen*: Opens the selected UserScreen in the Screens tab
- *Close UserScreen*: Closes the selected UserScreen

From the menu bar:



- *New UserScreen*: Creates a new UserScreen
- *Goto previous page*: Changes the current page to the previous one (if any) on the focused UserScreen
- *Goto Next Page*: Changes the current page to the next one on the focused UserScreen. If there is no next page, a new one is created
- *Goto Page Number ...*: Changes the current page to specific page on the focused UserScreen if it exists
- *Goto First Page*: Changes the current page to the first one on the focused UserScreen
- *Goto Last Page*: Changes the current page to the last one on the focused UserScreen
- *Move Objects ...*: Enables you to move one or several objects selected from one page to another. This command works not only in the current 'UserScreen' but also between various 'UserScreens' already created.

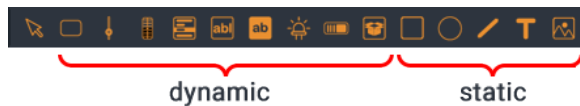


If you wish to move an object that has been 'linked' to start conditions or cue parameters, you MUST use this command. If you cut/copy/paste the objects from one UserScreen to another, all links to the objects are lost!

6.1.3 Graphical Objects

UserScreen Toolbar

The UserScreen Toolbar contains 2 types of graphical objects:



- Dynamic objects that can perform actions and have variable statuses.
- Static objects are decorative; they cannot be modified during runtime and do not trigger any actions in the system.

Dynamic Objects

There are two different types of dynamic objects: those primarily designed to perform actions, and those designed to monitor information. Advanced functionalities allow you to use these inversely, that is, monitoring with Action Objects and acting with Monitoring Objects.

- Button
- Slider
- Digital Slider
- List
- Text Edit
- Text Display
- LED
- Gauge
- Container

Static Objects

Static Objects are standard drawing objects that allow you to complement your Graphical Interface with information and design.

- Square/Circle/Line
- Static Text
- Image

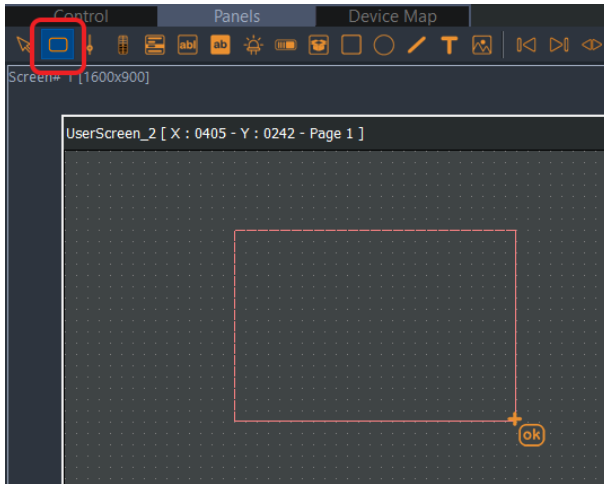


Before you can type in a different language than the one installed, you must configure the keyboard to match the language. See also [Regional and Language Options](#).

Creating Graphical Objects

Drawing an object

1. Use the UserScreen Toolbar to select the type of object.
2. Left click the mouse to define the top left-hand corner of the object.



3. Hold the left mouse button and drag the pointer to define the size of the object. Release the mouse button to complete the creation of the button.
4. Medialon Manager V7 generates an automatic name for the object. This name is based on object type and ID index. An object is identified in a project by its name. You can change this name in the Object Properties.

Properties	
Type	Button
Name	onoff_Button
Page	1
Left	70
Top	110
Width	250



Giving a detailed name to all your dynamic objects will help you during programming and debugging of your project.

6.1.4 Graphical Object Properties



marked properties only available in LocalPanel



marked properties only available in WebPanel



Some Graphical Object properties can contain an Expression to calculate the value, eliminating the need for a task for every Text Display, Gauge and LED.

Common Properties


Drawing Controls

Select the toolbar icon required, and drag out objects in a UserScreen. Constrain the aspect ratio by holding Ctrl while dragging. Make precise matching sizes and positions by using measures in properties and/or the alignment tools. Hold Shift while dragging to snap to other objects in the panel.

Position and Size

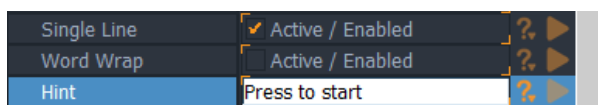
- *Left*: Defines the left position of the top left-hand corner of the Object relative to the UserScreen's top left corner
- *Top*: Defines the top position of the top left-hand corner of the Object relative to the UserScreen's top left corner
- *Width*: Defines the width of the graphical object
- *Height*: Defines the height of the graphical object

Design Properties

- *Colors*: Use the magic wand on the right side to open the *Color Selector Box*
- *Frame Width* and *Embossed Width*: Defines the number of pixels for the object's border
-  *Smooth Relief* option: makes embossed transitions smoother on Dynamic Objects.

Text Properties

- *Text Font*: Click on the magic wand to open the *Text Font* box. All fonts installed in the computer can be used in Medialon Manager.
- Text can be a single or multi line. If *Single Line* option is inactive, automatic carriage return can be activated with the *Word Wrap* option. Single Line and Word Wrap are mutually exclusive.




Use *Shift + Enter* key combination to insert a carriage return when both options are inactive.

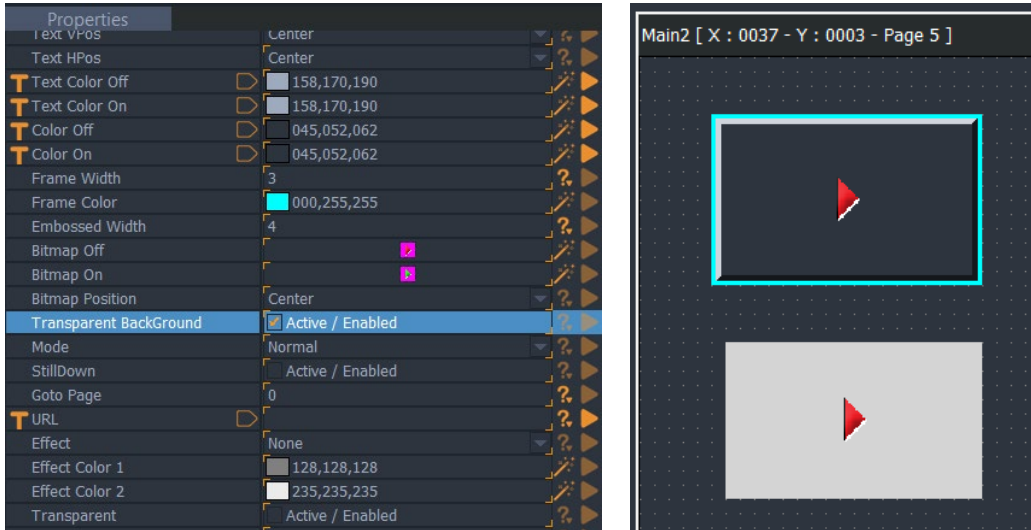
- *Hint*: A message to be displayed when the mouse hovers over the Object


Bitmaps

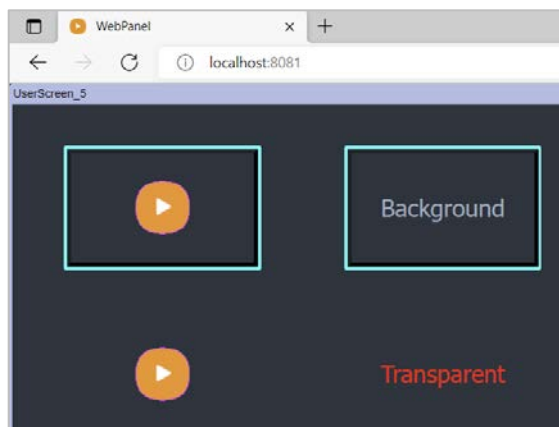
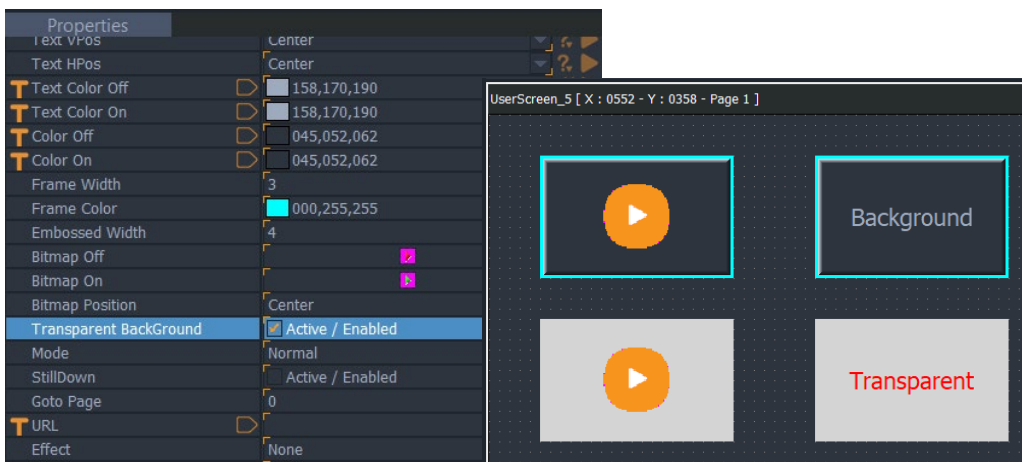
- Use the magic wand to open the *Image Selector* and define the bitmap.
- *Motif*: Defines the display of the Bitmap
- A Bitmap's position can be defined

Transparency

-  If your Bitmaps have a background color, make them transparent with the *Transparent Background* option. When active, all pixels in the bitmap with same color as the top left corner pixel are replaced by the object's color.




-  *Transparent BackGround* option makes the object transparent: text or bitmap are displayed without frame or background color. The transparency is only visible when rendered in the web browser.

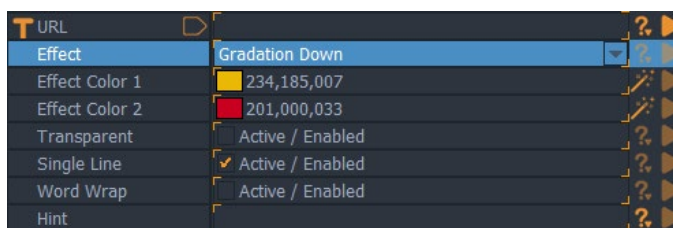
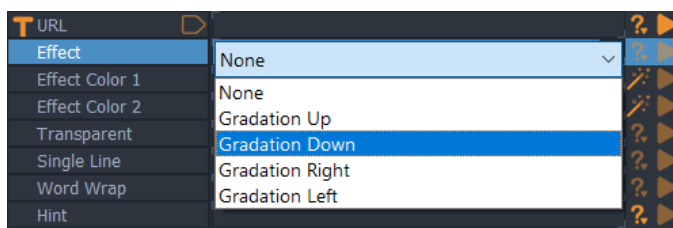


Visible and Enable



- *Visible*: whether the UserScreen object is visible or not (can be modified by the Object Control command of the Manager/Showmaster device).
- *Enable*: whether the UserScreen object is enabled for interactivity or not (can be modified by the Object Control command of the Manager/Showmaster device). Disabled objects do not react when clicked, though the objects' .Click variable does reflect the correct value.



Buttons

- *Status*: Button status (0 = Off; 1 = On)
- *Shape* option: changes the button from a rectangle to round
- *Text Position*: The vertical and horizontal positions of the text in the button can be set.
-  *Text Follow Bitmap* option makes sure the text does not overlay the image
- *Effect*: Instead of having one color per status, you can mix two colors and having a gradation effect.



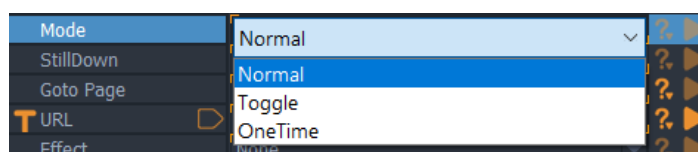
- *Transparent* option: makes the button invisible but active on the page. You can use this option to create hidden buttons or active zones on a bitmap.


  Vertical text position is locked to top when Word Wrap is enabled in WebPanel.

  Inside WebPanel, Effect properties cannot be drawn by the web browser if Bitmaps are used in the button.

Button Behavior

Buttons in Medialon Manager have three modes:



- *Normal*: Push button behavior, the button is On (1) while you press it. When you release it, its status goes to Off (0).
- *Toggle*: In this mode the button keeps its last status. You press it a first time, button status turns On (1). If you press it a second time, the status turns Off (0).
- *One Time*: In this mode, the user can press the button only once. You need to implement a command in a task to release a One Time button.
- *Still Down* option: draws the button in down position when its status variable is 1.
- *Goto Page*: this property switches the parent UserScreen to the specified page when the button is clicked.
-  *URL*: Opens another URL in the web browser containing the WebPanel. The new web page replaces the WebPanel where this button is pressed. If you wish to open the URL in a new browser tab, use this format for the URL property:
`javascript: window.open('http://www.sitename-example.com/');`



You may prefer to use Manager command 'Goto Page' inside tasks to switch pages.

Variable

Type	Name	Value	S	U
8	Main_Control.ConnectedWebPanels	0		
8	Main_Control.CtrlKeyStatus	0		
T	Main_Control.CurrentUserScreenName	Main		
8	Main_Control.KeyPress	77		
8	Main_Control.KeyStatus	0		
8	Main_Control.MouseClickLeft	1		
8	Main_Control.MouseClickRight	0		
8	Main_Control.MouseXPos	668		
8	Main_Control.MouseYPos	314		


- Name:[Button_name].Click
 - Type: Integer
 - Values: 1 while you click on it; 0 by default
- Name:[Button_name].Status
 - Type: Integer
 - Values: 0 = Off or Up; 1 = On or Down.

Sliders



Sliders are used to define levels. You need to set the minimum value (*Min Status*) and the maximum value (*Max Status*). The current position of the slider gives the current *Status*.

Max Status	255	
Min Status	0	
Status	255	

Select the 'slider' tool in the UserScreen toolbar and with the mouse draw your slider.

- *Ruler*: Defines the graduation marks and their color.
- *Slot*: Defines the width and the color of the slot.
- *Background Color*: Defines the color of the object; use the magic wand to select the color.
-  *Shape*: With this option you can create a rotary knob.
- *Effect* section: allows replacing the background color with a gradation between 2 colors

Variable


Lister				
Type	Name	Value	S	U
	Slider_66.Click	0		
	Slider_66.Status	47		

- Name:[Slider_name].Click
 - Type: Integer
 - Values: 1 while you click on it; 0 by default
- Name:[Slider_name].Status
 - Type: Integer
 - Values: Current Status between Min and Max Status.



Digital Sliders

Digital Sliders have similar properties to Sliders.

Select the 'digital slider' tool in the UserScreen toolbar and with the mouse draw your slider.

- *Ruler*: defines the graduation marks and their color.
- *Background Color*: defines the color of the object; use the magic wand to select the color.
-  *Shape*: With this option you can choose between a bar graph and a pie chart.

Variable

Lister				
Type	Name	Value	S	U
	DigitalSlider_67.Click	0		
	DigitalSlider_67.Status	54		

- Name:[Digitalslider_name].Click
 - Type: Integer
 - Values: 1 while you click on it; 0 by default

- Name:[Digitalslider_name].Status
 - Type: Integer
 - Values: Current Status between Min and Max Status. Editable through Object Properties.



Height	280	?	▶
Max Status	100	?	▶
Min Status	0	?	▶

Text Edits

User may need to enter text or numbers with the keyboard; the Text Edit object offers a field to do that. Select the 'Text Edit' tool in the UserScreen toolbar and with the mouse and draw your text field.

- *Text*: This property contains the result of keyboard typing.
- If you need to limit the length of typing, set the *Max Length* parameter. 0 means no limit.
- Text Edit can be used to enter a password; define a *Password Character* to display this character instead of the real entry.
- By default, a Text Edit is a string, you can force another *Edit Type*
 - None: any type of entry
 - Integer: only numeric values can be entered
 - Time: all entry will be formatted as a time value (hh:mm:ss/ff)
 - Date: valid value format as a date (ex: DD/MM/YYYY) – the order is dependent on the host PC's regional settings

Variable

Lister			
Type	Name	Value	S U
	Edit_70.Click	0	
	Edit_70.Text	Keyboard entry	

- Name:[Edit_name].Click
 - Type: Integer
 - Values: 1 while you click on it; 0 by default
- Name:[Edit_name].Text
 - Type: String
 - Value: result of keyboard typing



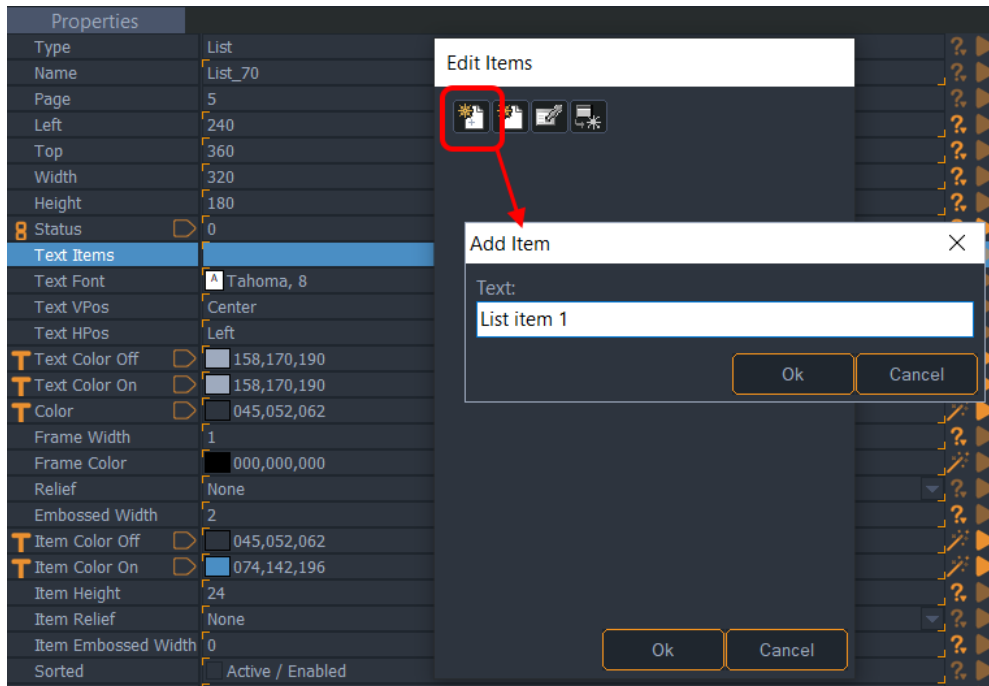
Vertical text position is top when Word Wrap is enabled and center when Single Line is active.

Lists

Medialon Manager offers a powerful tool: the list. The user can select an item from several items. See also [List Control](#).

Select the 'List' tool in the UserScreen toolbar and with the mouse draw your list.

- The position of the selected item in the list is given by the *Status* parameter (-1 = no selection, 0 = first item, 1 = second item, etc.).
- Text Items: Click on the magic wand to open the Edit Items box and add items to the list.



- *Text Position*: The position of the text in the edit field can be set
- *Sorted*: The list can be logically sorted. When Sorted is activated, two items in the list cannot have the same value.

Variables

Lister				
Type	Name	Value	S	U
8	List_70.Click	0		
8	List_70.Count	2		
T	List_70.SelectedText			
8	List_70.Status	-1		
T	List_70.Text	List item 1List item 2		


- Name:[List_name].Click
 - Type: Integer
 - Values: 1 while you click on it; 0 by default.

- Name:[List_name].Status
 - Type: Integer
 - Values: Selected index between 0 and List count; -1 means no selection.
- Name:[List_name].Text
 - Type: String
 - Value: list of all items, delimited by a line feed character after each item (hex 0A).
- Name:[List_name].SelectedText
 - Type: String
 - Value: Text of the selected index.
- Name:[List_name].Count
 - Type: Integer
 - Value: Count of list items.



Text Displays

Text information can be displayed in a UserScreen with a Text Display object.

Select the 'Text Display' tool in the UserScreen toolbar and with the mouse draw your display.



- *Text Position*: The position of the text in the display can be set.
-  *Inset*: Defines bitmap border size inserted in the display.
- *Motif Type*: Defines the way the bitmap is inserted in the Text Display
 - None
 - Stretch (force by default when Inset parameters are different than 0)
 - Pattern.

Variables

Lister			
Type	Name	Value	S U
	Edit_69.Click	0	
	Edit_69.Text	Display text	

- Name:[Textdisplay_name].Click
 - Type: Integer
 - Values: 1 while you click on it; 0 by default.
- Name:[Textdisplay_name].Text
 - Type: String

- Value: text to display.

  Vertical text position is locked to top when Word Wrap is enabled in WebPanel.

LEDs

Using default parameters, the color of the LED depends upon the value of its associated 'Status' property. A 'Status' value of 0 would set the color to black, a 'Status' value of 1 would set color to red, and so on.

You can change the color associated with each 'Status' value by editing the Color/State property in the Object Inspector. You can also use bitmaps instead of colors to reflect the LED's state by editing its bitmap/State property. Note that if a bitmap is associated to a state, this setting takes precedence over the color setting, though both are active. The bitmap is displayed on top of the background color.

At runtime, an LED Object can be programmed to blink automatically by setting the value of its associated 'Flash Frequency' and 'Sequence' variables, which appear in the variable Lister.

Frequency values are expressed in hundredths of a second. For example, setting the 'Flash Frequency' variable to 50 will change the 'Status' of the LED every half of a second when Manager is in Run Mode or Debug Mode. To disable LED blinking, just set its 'Flash Frequency' variable to 0.

The 'Sequence' specifies the sequence of states that would be traversed by the LED if flash mode is on. Each state's number must be separated by a non-numerical symbol. For example, setting the value of the 'Sequence' variable to '1,3,10' will automatically change the state of the LED from 1, to 3, then to 10, then to 1 and so on.

Select the 'LED' tool in the UserScreen toolbar and with the mouse, draw your LED (use Ctrl+drag to maintain aspect ratio).

Properties

- *Status*: Current status of the LED from 0 to 255
- *Shape*: Shape of the LED (round or rectangle)
- *Color*: Indicates the color per status of the LED (0-255)
- *Bitmap*: Allows you to insert one bitmap for each status of the LED (0-255)
- *Frequency*: Speed of flashing LED; the higher the number, the longer the interval.
- *Sequence*: defines the pattern of statuses the LED will run through when the Frequency is other than 0. Use a non-numerical symbol to separate the different status values within the pattern.

Example: 0,1,2,3,2,1

Variables

Lister				
Type	Name	Value	S	U
B	Led_71.Click	0		
B	Led_71.FlashFrequency	15		
T	Led_71.Sequence			
B	Led_71.Status	1		

- Name:[LED_name].Click
 - Type: Integer
 - Values: 1 while you click on it; 0 by default.
- Name:[LED_name].Status
 - Type: Integer
 - Values: Current status of the LED from 0 to 255



LED status is not refreshed when FlashFrequency and Sequence are used and the LED is not shown

- Name:[LED_name].FlashFrequency
 - Type: Integer
 - Values: 0=No Flash; 1=very fast; greater value = slower flash speed
- Name:[LED_name].Sequence
 - Type: String
 - Value: Pattern of statuses the LED will run through when the Flash Frequency is other than 0. Use a non-numerical symbol to separate the different status values within the pattern.

Example: 0,1,2,3,2,1

Gauges

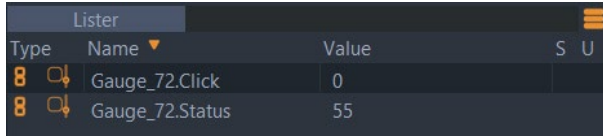
The gauge object can be used to display a level or a ratio like a progress bar.

Select the 'Gauge' tool in the UserScreen toolbar and with the mouse draw your gauge.

- *Status*: Current status of the Gauge.
- *Min/Max Status*: defines the Min and Max values of the Gauge status.
- *Direction*: defines in which direction the gauge will move (Right, Left, Up or Down).
- *Color Off*: Defines the color of the background of the gauge; use the magic wand to select the color.

- *Color On*: Defines the color of the filled portion of the gauge; use the magic wand to select the color.
- *Text Color*: Defines the color of the text visible at the center of the gauge; use the magic wand to select the color. *Hide Text* option allows to get a gauge without value indicator.

Variable



Type	Name	Value	S	U
8	Gauge_72.Click	0		
8	Gauge_72.Status	55		

- Name:[Gauge_name].Click
 - Type: Integer
 - Values: 1 while you click on it; 0 by default
- Name:[Gauge_name].Status
 - Type: Integer
 - Values: Current Status between Min and Max Status.



Containers

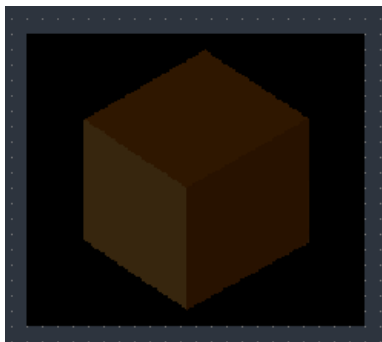
Containers are a very specific active device.

Select the 'Container' tool in the UserScreen toolbar and with the mouse draw your container.

Local Panel Containers actively display Device Control Panels, as well as MXM specific information, such as Database, Video or a Web Browser

WebPanel Containers show iFrame enabled websites including projector web interfaces, live video feeds from cameras or media servers.

Other than the standard graphical configuration properties, containers do not have any specific properties or any variables. They are in fact 'windows' that are used to display the information or media in a specific and controlled location.



To set the content of a container, you must use the Manager 'Device Control' Command.

Static Graphical Objects

Static graphical objects can help you to design nice user interface.

Square

Select the 'Square' tool in the UserScreen toolbar and with the mouse draw your square. Square can be filled with a color or an effect color.

Circle

Select the 'Circle' tool in the UserScreen toolbar and with the mouse draw your circle. Circle can be filled with a color or an effect color.

Line

Select the 'Line' tool in the UserScreen toolbar and with the mouse draw your line. Color and size can be adjusted.

Static Text

Select the 'Static Text' tool in the UserScreen toolbar and with the mouse draw your text area where you want to insert text like a title or label.

Image

Select the 'Image' tool in the UserScreen toolbar and with the mouse draw your Image area.

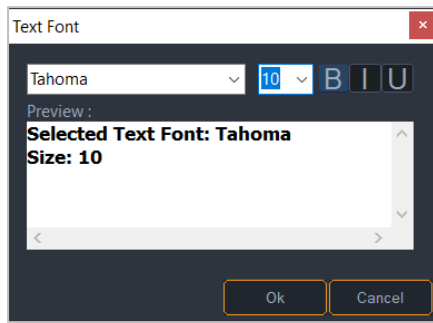
- *Motif Type*: Defines the way the bitmap is inserted
 - None
 - Stretch
 - Pattern
 - Size To Bitmap: When enabled, Image area is resized to inserted bitmap size.

6.1.5 Graphical Object Tools

Graphical Object Tools are accessible with the properties wizard button .

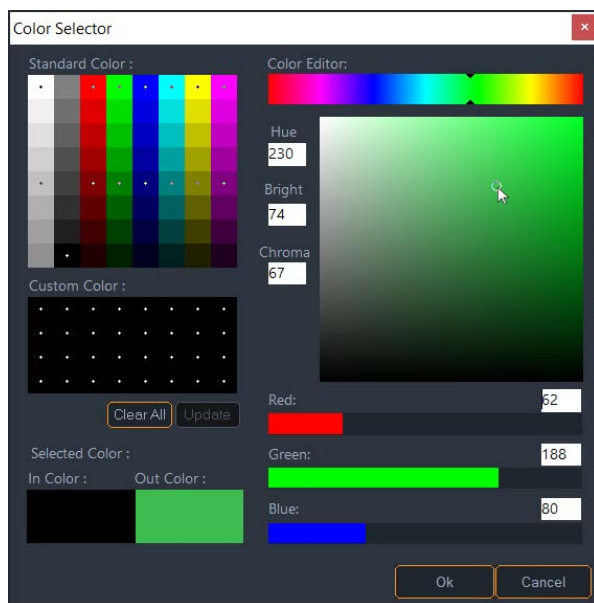
Font Selector

Use the dropdown lists to edit the style and size of the graphical objects text properties.

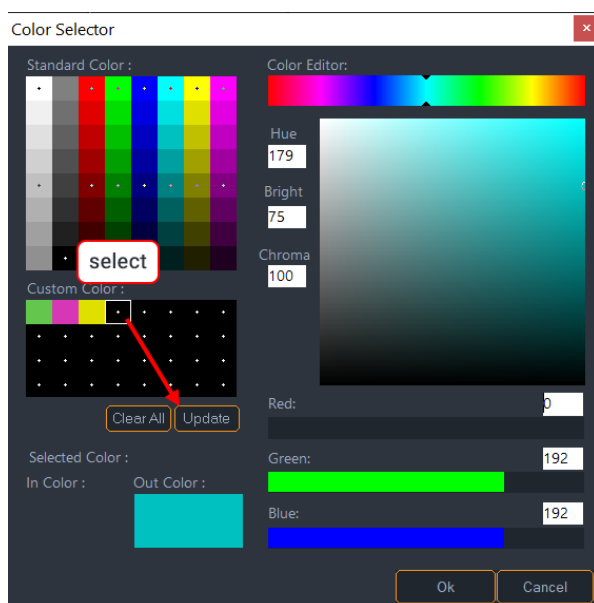


Color Selector

Allows you to edit and store the various color properties of the selected object.



To define your color as a new custom color, select a custom block and click the Update button.



i You can also sample custom colors at any time when you add graphical objects. Without opening the picker, use the keyboard shortcut Ctrl + Alt and click on the displayed color. When you next open the color picker you will see the sampled color added to the custom color list.

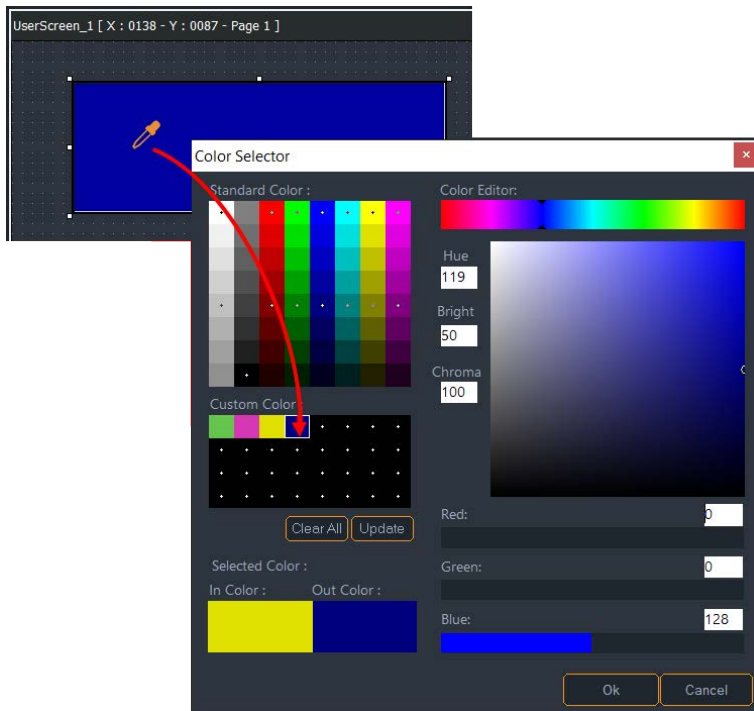
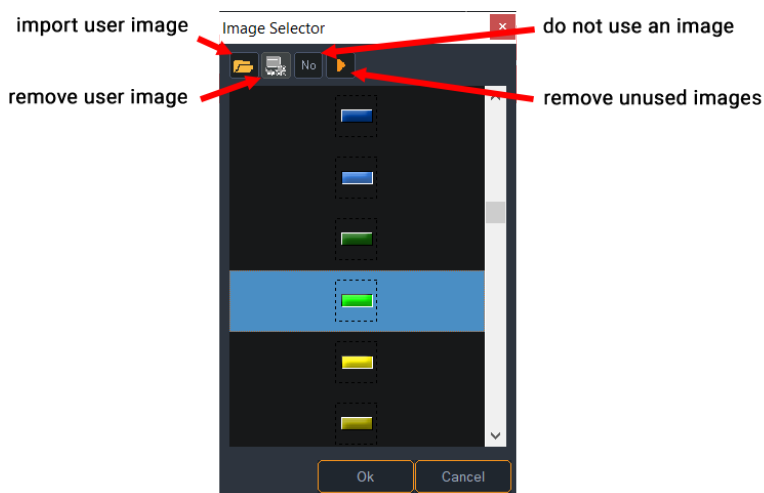


Image Selector

The Image selector lists the images that can be used as object properties. You may select from the predefined images available or import your own.



Load Image

Use this command to import images in your project. A file browser allows you selecting any image on your PC. A copy of this image is added to the ProjectName.ImageFolder.

i Note that Medialon Manager V7 can accept .bmp, .jpg, .png and .gif format.

Remove Image

If an imported image is unused, you can remove it from the project Image folder. It is not removed automatically.



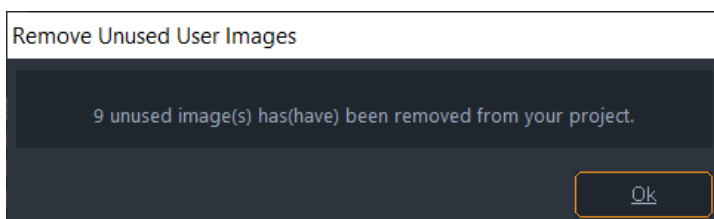
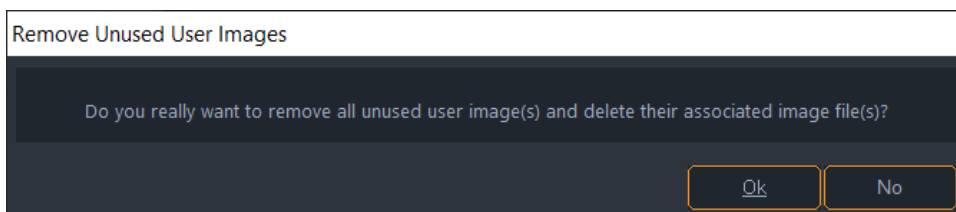
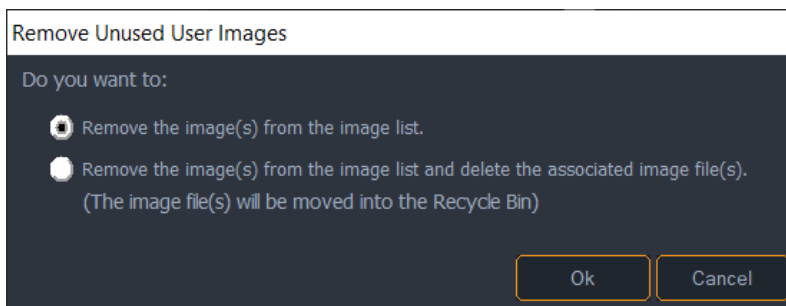
Built-in images cannot be removed.

No Image

The 'No Image' button removes the image from the object.

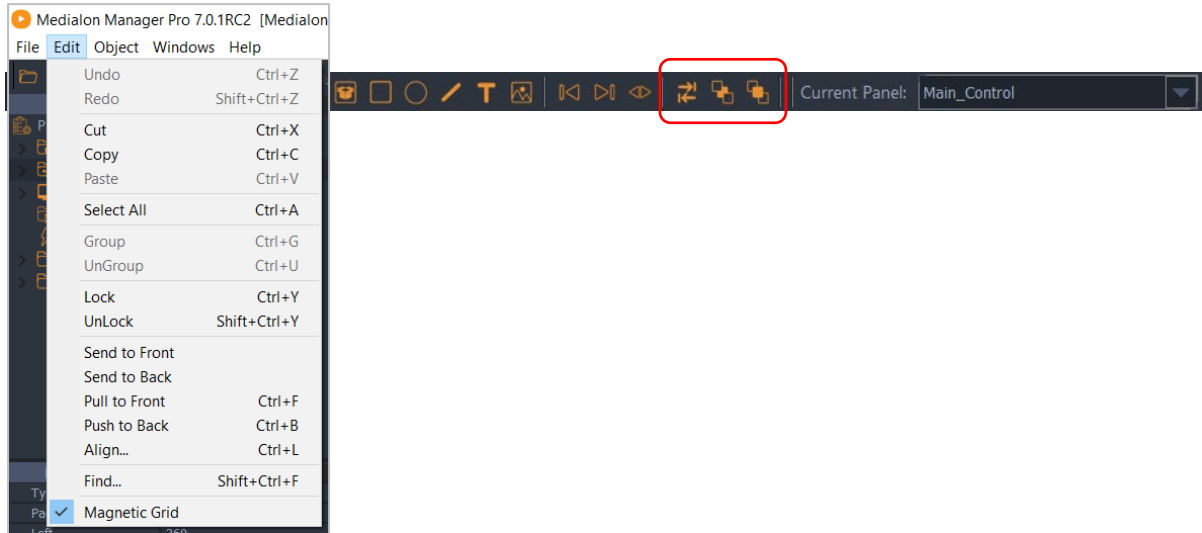
Remove unused image(s)

During the evolution of your project, you may have several images not used, you can clean your project and the image folder with this menu.



Layer and Position Tools

Layer and Position tools are available in the Panel editor toolbar or in the Edit menu.



Send To Front

Brings the selected Graphic Objects to the foreground.

Send To Back

Pushes the selected Graphic Objects to the background.

Pull To Front (Ctrl+F)

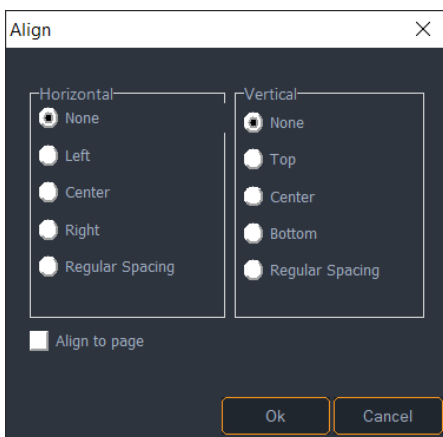
Pulls the selected Graphic Objects (shapes only) one level towards foreground.

Push To Back (Ctrl+B)

Pushes the selected Graphic Objects (shapes only) one level towards background.

Align... (Ctrl+L)

Opens the Align dialog box for aligning selected Graphic Objects.




Align to Page aligns selected objects to the UserScreen page's extents (left/center/right).

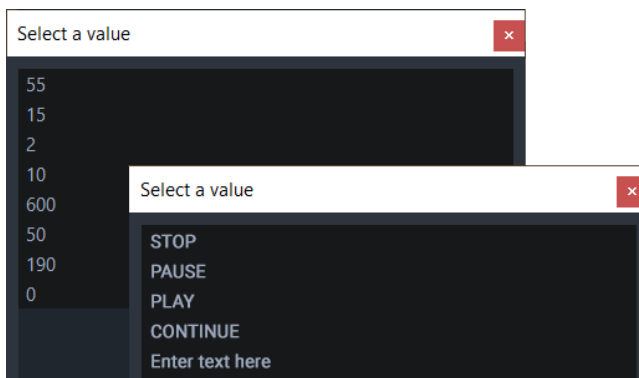
Magnetic grid

Turns the UserScreen Magnetic Grid On/Off.

In the 'UserScreens', the Magnetic Grid enables you to position objects in relation to each other at distances defined by the virtual grid. This grid is shown by points on the 'UserScreens'. When the Magnetic Grid is turned off, objects can be moved freely without snapping to grid points.

Value Selection Window

When you need to type the same information in Property fields of different objects several times, you may find it useful to use the value selection window . Every time a numerical or alphabetical field is filled out, the values are stored in this window. By clicking on the 'Select a Value Icon', you can access this list and insert the value without typing. The values are stored in a separate window whether they are numerical or alphabetical value; you cannot have access to the numerical values from an alphabetically arranged property field.



6.2 Variables

6.2.1 Defining Variables

Variables are objects that store a value. They are expressed as alphanumeric names that can be used for calculations and the execution of tasks. The value changes according to the state of an object, the result of a calculation, a user's action or the information sent back from a device.

Variables may characterize the states or actions of 'Devices' and objects present in the 'UserScreens'.

Independently of the variables associated with 'Devices' and 'UserScreens', 'User Variables' may be created to suit your programming necessities.

Categories

System

System variables are variables that pertain to Manager and to other 'Devices'. They cannot be deleted (except for Medialon On Network variables). The Manager variables concern the date, time, states and actions of the keyboard and the mouse.

Screen

Every Panel returns information about mouse and keyboard activity via screen variables. Each 'UserScreen' is associated with a variable (its page numbers). Each active object also generates one or several variables. The variables linked to the 'UserScreens' cannot be deleted. They disappear only when the object or 'UserScreen' that created them is deleted.

User

User variables are variables created by the programmer. They can be easily modified or deleted.

Types of Variables

Integer

Integer variables express a whole number, characterizing the state of an object, the X/Y position of the mouse pointer, the value of a slider, or a visitor count, for example. The initial value must be indicated. The default value is zero. The 'Min value' and 'Max value' fields indicate, by default, the minimum and maximum values that the variable can take. The values of these fields can be used for programming, calculations and displays.

Real

Real variables are like integer variables, with the additional advantage of having floating-point representation (decimal values). The precision of the display can be set to the desired number of decimal places (Medialon Manager rounds off to the closest value).

String

String variables are a chain of characters that can be displayed as such or integrated into a sentence. Such a chain of characters is displayed with the typographical choices (font, size, color) attributed to its display system.

Enum

Enum variables are variables that list pre-determined values: periodical information or days of the week, for example. Enums have both an integer value (first item is 0, second item is 1, etc.) as well as a string value.

Time

Time variables allow various modes for dividing and displaying time, on the basis of the frame rate or timecode standard (24, 25 or 30 frames per second) and a decimal time-splitting of seconds (1/100 and 1/1000 second).

Date

Date variables contains a date. This may be used for date calculations, comparisons, or programming periodical, exceptional or random events. Medialon Manager V7 follows the PC's regional options to display dates.



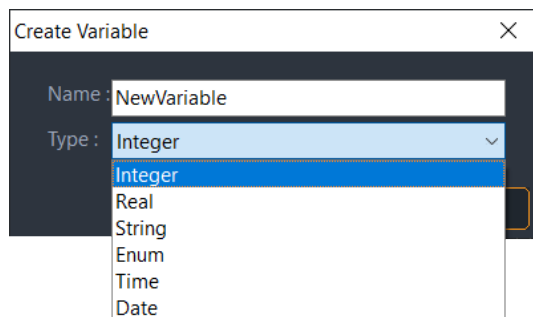
Date format must be a complete format, e.g. DD/MM/YYYY, YYYY-MM-DD or MM.DD.YYYY. '8/4/07' is not recognized by Medialon Manager.

Persistent Variable

A Persistent variable has its value stored to disk, so when Medialon Manager is switched off or even after an unforeseen power cut, the value is restored the next time the project runs.

6.2.2 Creating User Variables

You can create a variable either by right-clicking on the 'User Variables' Object in the Browser tab, right-clicking in the Lister, or by using the keyboard shortcut *Shift+Ins*.



The type of variable must be chosen and the variable must be given a name that has not already been used and which does not include a reserved keyword or characters used for calculations (/ +: -, etc.).

Variable names may not begin with numbers.

Examples:

Not valid: on/off Valid: on_off
 Not valid: 2_up Valid: Up_2

When variables are created, Medialon Manager proposes, by default, an indexed version of the previously entered variable name and maintains the same type as the previously created

variable. In the same way, when a variable is duplicated, the new variable takes the name [name of the variable copied]2.

6.2.3 Variable Properties

Just like any other Object in Medialon Manager V7, variables have properties. If you wish to edit them, you must take into account the category and the type, as some elements are not modifiable.

Variable Names

Variables are referenced in the system by their names. System and Screen variables are automatically named using a logical procedure. The names are divided into two parts.

[ObjectName].[Description]

Example: Manager.CurrentTime or Button.Status

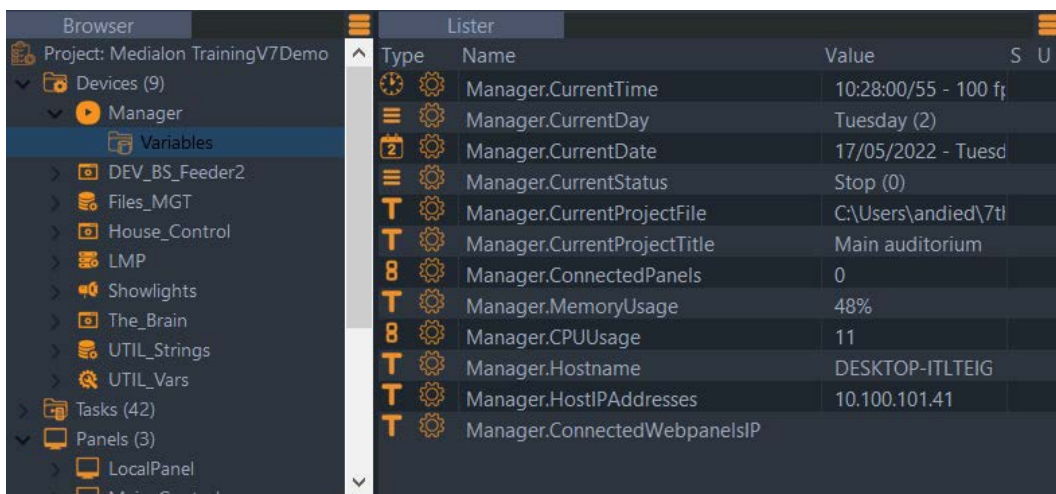
You may choose to follow this nomenclature when creating user variables or to use another one. However, it is very important to give a precise name to each variable you create, this will ease navigation when searching for the variable during programming.

System Variable Properties

As we have seen, System variables are created with the devices. Manager, being seen as a device automatically generated in the project, will create Manager system variables. Each device created has its own variables, and so do tasks if you chose to create task variables when creating the task.

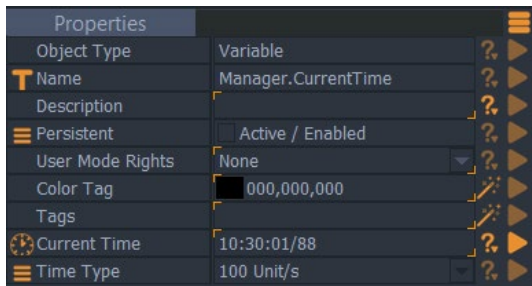
To access the list of an object's variables, select the variables folder of the desired device or task and the Lister updates to display them.

Lister columns may be sorted by clicking on the column heading.



Type	Name	Value	S	U
🕒	Manager.CurrentTime	10:28:00/55 - 100 ft		
📅	Manager.CurrentDay	Tuesday (2)		
📅	Manager.CurrentDate	17/05/2022 - Tuesd		
⚙️	Manager.CurrentStatus	Stop (0)		
📁	Manager.CurrentProjectFile	C:\Users\andied\7ti		
📁	Manager.CurrentProjectTitle	Main auditorium		
📊	Manager.ConnectedPanels	0		
📊	Manager.MemoryUsage	48%		
📊	Manager.CPUUsage	11		
🌐	Manager.Hostname	DESKTOP-ITLTEIG		
🌐	Manager.HostIPAddresses	10.100.101.41		
🌐	Manager.ConnectedWebpanelsIP			

To display a variable's properties, select it to update the Properties tab.



System variables are linked to status and information of the devices that are active within the session, therefore, not all parameters are editable. System variables cannot be Persistent since their parameters are linked to the actual status of the devices, not the last state they were in when the project was shut down.

The editable properties of system variables are marked with orange corners.

User Mode Rights defines if this variable is visible in the User Mode.

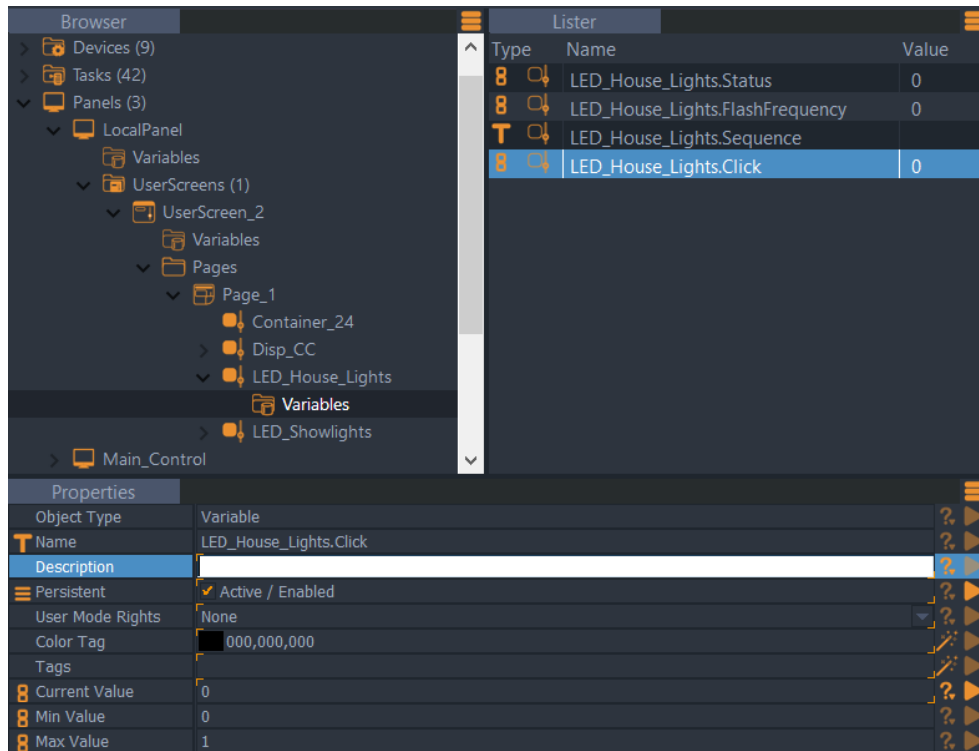
Screen Variable Properties

All screen Objects created in your project, whether Panels, UserScreens or Graphical Objects, also have variables linked to them. To access the variable Properties of the Screen Objects, you must browse to the correct UserScreen or Page and open the variables folders in the Lister tab.

Please note that the graphical Object Properties and the Graphical Object's Variable Properties are two different and separate properties.

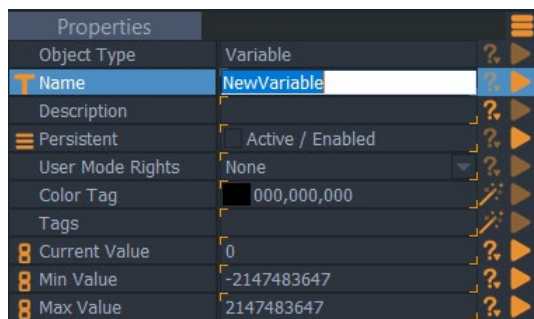
User Mode Rights defines if this variable is visible in the User Mode.

Screen variables can be set to persistent, which causes the value to be stored to disk each time they change value; the values are restored each time the project is loaded and runs.



User Variable Properties

When you create a user variable, there is no direct link to any object in your project, therefore all the properties are editable, including the name and values.



User Mode Rights defines if this variable is visible in the User Mode.

User variables can be set to persistent, allowing you to automatically retrieve the last state they were in when the project was last closed.

6.2.4 Using Variables as Cue Parameters

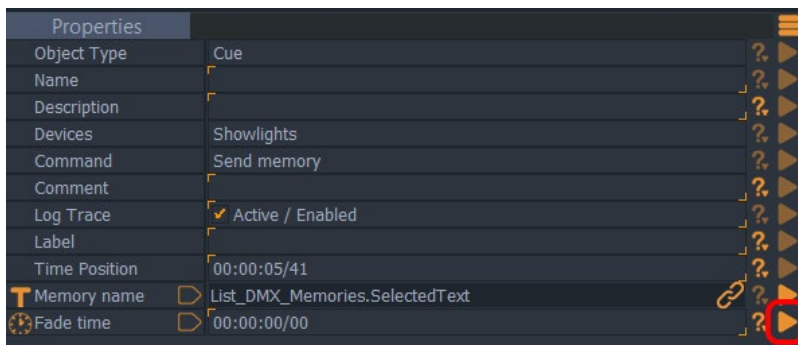
Variables may be used as parameters for command cues to dynamically control one or all the parameters. This useful function allows you to dynamically determine the parameters of a command at the time it is sent.


Automation Programming

All editable parameters of command cues can use variables as their value. Medialon Manager does not validate any of these values, so the variable that is used MUST have a value that is interpretable by the recipient.

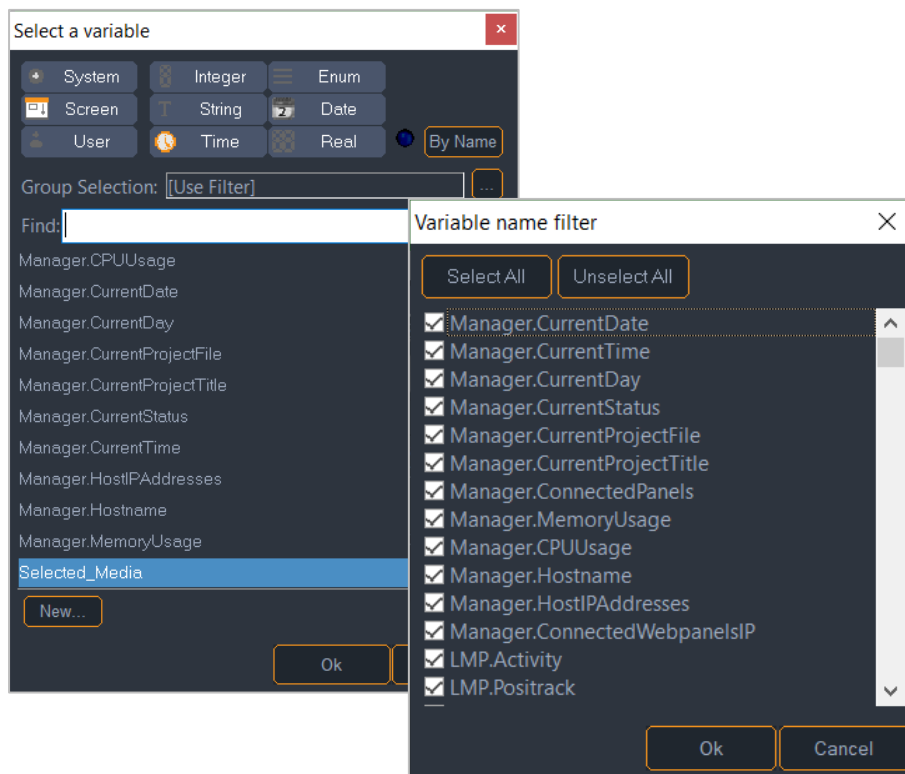
For example, a 'Locate to Time' Command is only valid if the variable is a 'Time', a 'Time formatted String' or an integer (Integers are translated to Time, based on the Frame Rate of the Time parameter; if the frame rate is 1/100s, and the integer 100, the corresponding time value will be '00:00:01/00').

To use a variable as a parameter for the command cue, select the command cue you want to configure and click on the 'arrow' to the right of the desired command parameter.

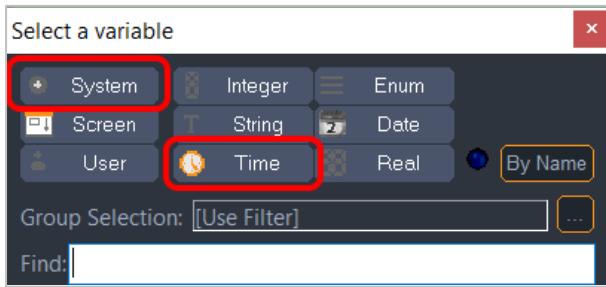


Note: 'Memory Name' has a link icon , which indicates that this variable is linked to another.

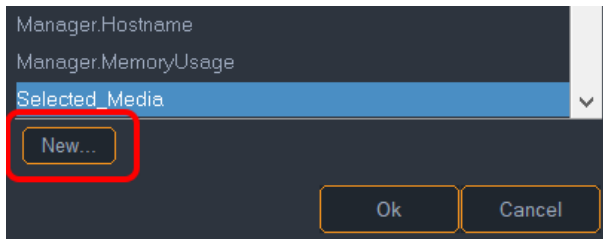
This opens the Variable Selection Window where you can search for and select the variable to use. Filter by Type (top) or by User Group or 'By Name':



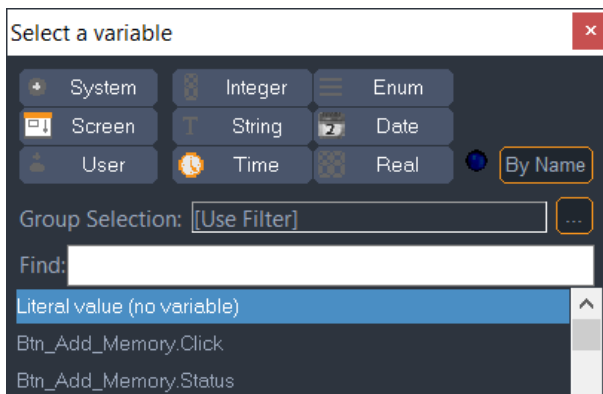
You can also use Category and Type buttons to show or hide variables.



Use the New... button to create a new user variable directly from the variable selector.



If you decide not to use a variable as a parameter, select 'Literal value (no variable)' at the top of the variable list.



6.3 Expressions

6.3.1 Logical Expressions

Logical Expressions provide the normal functions of computer programming, using Boolean operators. An expression is a mathematical (or logical) string; it contains variables, constants and operators (+, -, *, /, (,), etc.). Expressions are evaluated using operators between variables and constants to provide a result. The result could be used as a logical result (true or false) or could be stored into a variable for future use.

Operators

Several operators are available and can be divided in two main categories: arithmetical operators and logical operators. Operators have priority among them.

$2 + 4 * 3$ equals to 14 (not 18) because $*$ operator has a higher priority than $+$ operator.

The priority of operators is: $^$, $-$ (Neg), $\%$, $*$, $/$, $+$, $-$, $==$, $!=$, $>$, $<$, $>=$, $<=$, $|$, $\&$, $=$, $\#$.



Arithmetical operators

- $+$ Add a value to another one
- $-$ Subtract a value from another one or use as a negative operator when there is only one value
- $*$ Multiply a value by another one
- $/$ Divide a value by another one
- $\%$ Modulo; returns the remainder of the division from the first value by the second
- $^$ Power operator
- $=$ Assignment operator; set the first value to the contents of the second

Logical (Boolean) operators

- $==$ Equality comparison operator; compares the contents of two values. If they are equal the result is true (1) otherwise it is false (0).
- $!=$ Difference comparison operator; compares the contents of two values. If they are equal the result is false (0) otherwise it is true (1).
- $>$ Greater than operator
- $<$ Less than operator
- $>=$ Greater than or equal to operator
- $<=$ Less than or equal to operator
- $|$ OR operator; one or the other value must be true to evaluate as true
- $\&$ AND operator; both parts of the Expression must be true to evaluate as true

Change operator; used only with variables, the content of a variable is compared to its previous value to define if it has changed

() Parenthesis symbols are used to force priorities between operators

Evaluation expressions

Evaluation expressions are either true or false. Those expressions are used for conditions (i.e. If/Then/Else, While/EndWhile, WaitFor, etc.):

```
Button_StartShow_1.Status == 1
```

If Button_StartShow_1.Status is equal to 1, the expression is true and the command is executed.

Assignment expressions

Assignment expressions are used to store a value into a variable:

```
ShowSelection = 2  
TimeToShow = "01:00:05/00"
```

ShowSelection variable is set to 2 whereas TimeToShow variable is set to the time '01:00:05/00'.

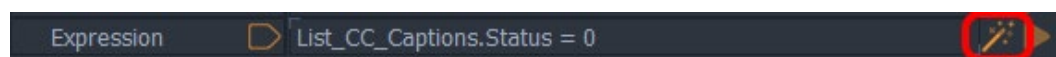
Both evaluation and assignment expressions can be combined:

```
ShowSelection = (Slider_SelectShow.Status == 5)
```

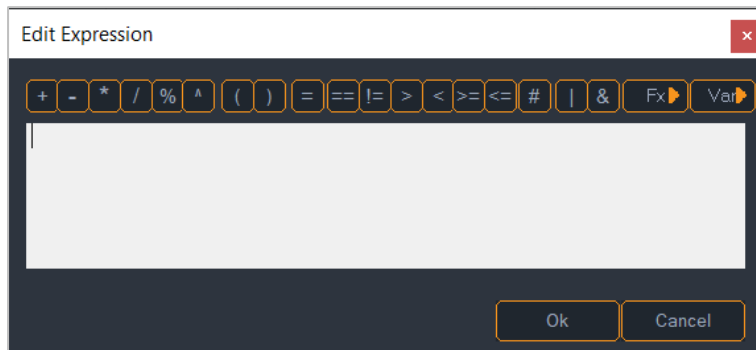
If `Slider_SelectShow.Status` is equal to 5, the expression in the parenthesis is true and therefore the variable ShowSelection is set to 1 (true is considered as 1), otherwise if `Slider_SelectShow.Status` is equal to 0, the expression in the parenthesis is false, therefore the ShowSelection variable is set to 0 (false is considered as 0).

6.3.2 The Expression Window

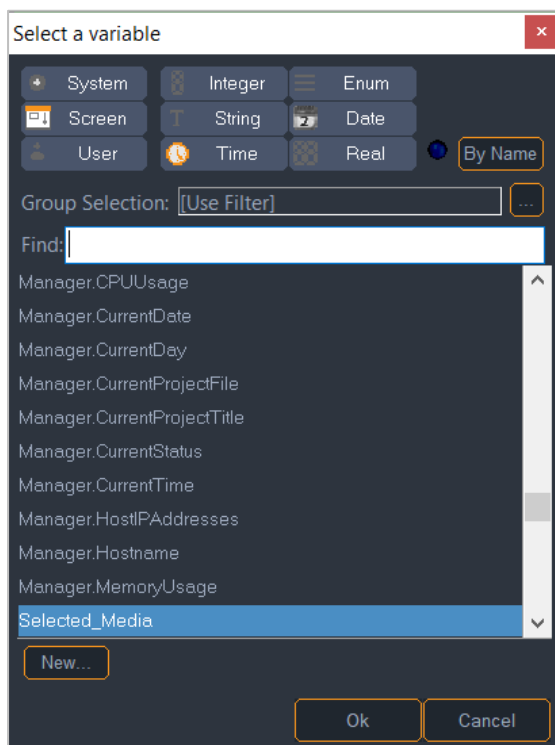
When you create an expression cue within a task, you need to write the expression out, so the calculation is performed. To access the 'Edit Expression' window, select the cue and click on the 'Magic Wand' of the Expression Property to open the Edit Expression Window.



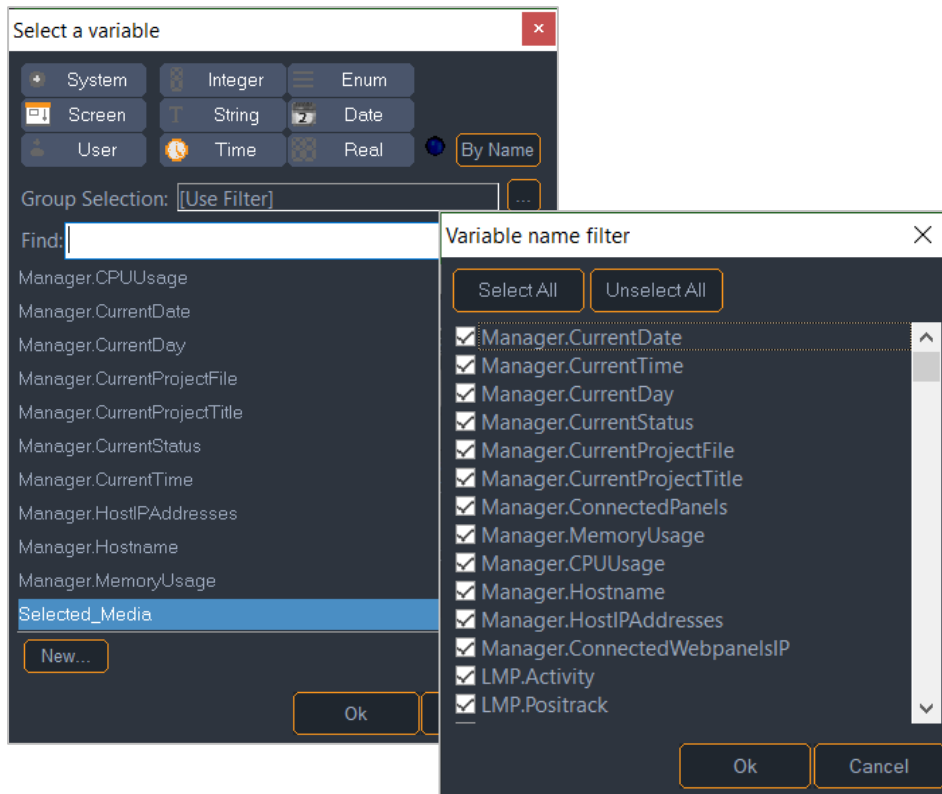
Alternatively, you can double-click on the expression field in the Properties Tab.



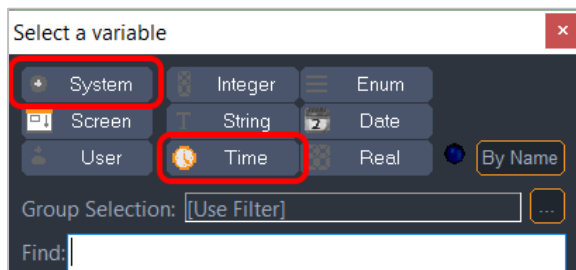
To create Expressions, you will need to choose variables to use. Simply click on the variable selection icon (Var button) to open the Variable Selection Window. All the filters are active when you first open this window. If you do not wish to display the variables of one category or type, click on the Icon and they are filtered out.



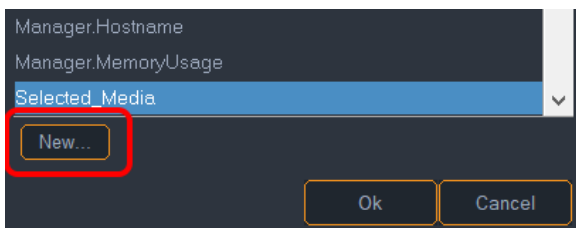
Filter by Type (top) or by User Group or 'By Name':



You can also use Category and Type buttons to show or hide variables.



Use the New... button to create a new user variable directly from the variable selector.



When you select a variable from the list, the variable appears in the 'Edit Expression' as it is named in your project.

You will also need to select the operators that will perform the calculation you require. Select the operators by clicking once on their icon; they are inserted automatically in the Expression.

Expressions can also be entered on the keyboard as long as the syntax is scrupulously respected.



All string, time, and date values must be enclosed within quotes: 'textvalue', '00:01:25/15', '08/07/2007'.

6.3.3 Operations

Conversion Rules

In an operation, the type of the result is the type of the left most operand of the operation. The other operand is converted before the operation.

Table 1: Conversion results

Left/Right	Integer	Real	String	Time	Date	Enum
Integer	Integer	Integer	Integer	Integer	Integer	Integer
Real	Real	Real	Real	Real	Real	Real
String	String (1)	String (1)	String (2)	String (2)	String (2)	String (1)
Time	Time (3)	Time (3)	Time (3)(5)	Time (3)		Time (3)
Date	Date (4)	Date (4)	Date (4)(6)		Date (4)	Date (2)
Enum	Enum	Enum				Enum

Operation restrictions:

1. Addition, Subtract, Multiply, Modulo (Remainder)
2. Addition only
3. Addition, Subtract, Multiply, Divide
4. Addition, Subtract
5. String must have a Time format '00:00:00/00'
6. String must have a Date format (for example: '00/00/0000'). See also [Date](#).



To convert an Integer that ranges from 0 to 100 to a Real that ranges from 0.0 to 1.0, use this kind of expression: $\text{Float} = (0.0 + \text{Integer}) / 100$.

The right side of the operator is converted to the type of the leftmost item before evaluation of the expression.

Examples:

String = Integer

(Edit.Text = 10)

The Integer (10) is first converted into the String ('10'), then stored to the Variable (Edit.Text).

Date >= String

(Manager.CurrentDate >= '01/04/2010')

The String ('01/04/2010') is converted into a Date before being compared to Manager.CurrentDate.

Operations on Strings

Arithmetical operators can be used to perform some kinds of string management:

- Truncates the string from the right: (String = String - 4): removes the last 4 characters from String. If String had the value 'ABCDEFGH', then the new value is 'ABCD'.
- + Concatenate two strings: (String = String + 'enation'): adds the sub string 'enation' to the contents of String. If String had the value 'Concat', then the new value is 'Concatenation'.
- * Concatenate a string multiple times: (String = String * 3): Concatenates the contents of String 3 times. If String had the value 'W', the new value is 'WWW'.
- % Truncates the string from the left: (String = String % 3): Keeps the last 3 characters of String. If String had the value 'ABCDEFGH', then the new value is 'FGH'.

Converting Time

Time can be converted into an integer by using an expression like

(MyInteger = Manager.CurrentTime) MyInteger contains a value which represents the number of frames, based on the frame rate of the time variable.

Time can be converted into a string by using an expression like

(MyString = "" + Manager.CurrentTime) MyString contains 'HH:MM:SS/FF' value of the time.

Comparing Time

To compare two time values, it seems logical to use the == operator. But because expressions are evaluated one after another, it is plausible that expression:

(Manager.CurrentTime == '01:00:05/00')

may not evaluate as true when current time is '01:00:05/00' because the expression was evaluated just before or just after the computer's clock reached the target time.

It is better to create a time window to compare time values. The expression

((Manager.CurrentTime >= '01:00:04/90') & (Manager.CurrentTime <= '01:00:05/10'))

is better and guarantees that the condition will evaluate as true. The width of the time window must be adjusted according to the required precision.

Formatting Time Display

To remove frames or hours when you want to display a time, you need first to convert the expression as a string operation and use truncate features.

- `(TextDisplay = "" + TimeVariable - 3)` gives HH:MM:SS as result.
- `(TextDisplay = "" + TimeVariable %8)` gives MM:SS/FF as result.

Calculating and Converting Dates

Dates can be converted into integers by using an expression like `MyInteger = Manager.CurrentDate`. `MyInteger` contains a value which represents the number of days elapsed since the reference date: 01/01/1900. `MyDate = 38312` causes the variable `MyDate` to take the value 11/23/2004 because 38312 is the number of days elapsed since the reference date (01/01/1900).

This mechanism can be used to calculate a new date based on a known date and a number of days. For instance, we may want to calculate what date is 12 days from today: `MyDate = Manager.CurrentDate + 12` causes `MyDate` to take the value of today's date plus 12 days.

Creating a Toggle Value

It is often useful to have a value which alternatively swaps between two values (for instance 0 and 1). This could be done using `If/Then/Else` commands, but it is faster and simpler to use an expression to do it:

```
ToggleValue = 1 - ToggleValue
```

This expression sets the value of the variable `ToggleValue` to 1 when its value is 0 and 0 when its value is 1, each time the expression is executed.

Creating a value selector with buttons

When a choice must be made between options using buttons, for instance choosing between 5 options, an expression can be used instead of a complex task that evaluates each button status to determine the selection.

After creating 5 buttons named `Button_1` to `Button_5`, those buttons are programmed to start the same task with the following expression:

```
((Button_1.Status==1) & (Button_1.Status#)) |  
((Button_2.Status==1) & (Button_2.Status#)) |  
((Button_3.Status==1) & (Button_3.Status#)) |  
((Button_4.Status==1) & (Button_4.Status#)) |  
(Button_5.Status==1) & (Button_5.Status#)
```

Once the task is started by one of those buttons, the goal is to know which button has started the task. The expression:

```
Selection = (Button_1.Status == 1) * 1 +
(Button_2.Status == 1) * 2 +
(Button_3.Status == 1) * 3 +
(Button_4.Status == 1) * 4 +
(Button_5.Status == 1) * 5
```

Selection is set to a value from 1 to 5 depending on which button started the task. Only one button can start the task at a time, so when a button status has the value 1, all other button statuses have the value 0. This expression can be evaluated as follows:

```
Selection = 1 * 1 + 0 * 2 + 0 * 3 + 0 * 4 + 0 * 5 = 1 when Button_1 is down.
Selection = 0 * 1 + 1 * 2 + 0 * 3 + 0 * 4 + 0 * 5 = 2 when Button_2 is down.
Selection = 0 * 1 + 0 * 2 + 1 * 3 + 0 * 4 + 0 * 5 = 3 when Button_3 is down.
Selection = 0 * 1 + 0 * 2 + 0 * 3 + 1 * 4 + 0 * 5 = 4 when Button_4 is down.
Selection = 0 * 1 + 0 * 2 + 0 * 3 + 0 * 4 + 1 * 5 = 5 when Button_5 is down.
```

6.3.4 Function Operators

Medialon Manager V7 integrates Function Operators allowing you to extract information from complex variables or create 'equations' using variables.

Hours(TIMEVAR)

Extracts the [Hours] value from a time variable

- Input: TIMEVAR: time variable
- Output: Integer corresponding to hours

Example:

```
TimeVariable = "12:34:15/30"
Hours(TimeVariable) = 12
```

Minutes(TIMEVAR)

Extracts the [Minutes] value from a time variable

- Input: TIMEVAR: time variable
- Output: Integer corresponding to minutes

Example:

```
TimeVariable = "12:34:15/30"
Minutes(TimeVariable) = 34
```

Seconds(TIMEVAR)

- Extracts the [Seconds] value from a time variable

- Input: TIMEVAR: time variable
- Output: Integer corresponding to seconds

Example:

```
TimeVariable = "12:34:15/30"  
Seconds(TimeVariable) = 15
```

Frames(TIMEVAR)

Extracts the [Frames] value from a time variable

- Input: TIMEVAR: time variable
- Output: Integer corresponding to frames

Example:

```
TimeVariable = "12:34:15/30"  
Frames(TimeVariable) = 30
```

Day(DATEVAR)

Extracts the [Day] value from a date variable

- Input: DATEVAR: date variable
- Output: Integer corresponding to day of DATE

Example:

```
DateVariable = 01/18/2020  
Day(DateVariable) = 18
```

DayOfTheWeekAsName(DATEVAR)

Extracts the [Day_Name] value from a date variable

- Input: DATEVAR: date variable
- Output: String corresponding to the day's name

Example:

```
DateVariable = 01/18/2020  
Day(DateVariable) = "Saturday"
```

DayOfTheWeekAsNumber(DATEVAR)

Extracts the [Day_Number] value from a date variable

- Input: DATEVAR: date variable
- Output: Integer corresponding to the day's number in the week (Sunday=1)

Example:

```
DateVariable = 01/18/2020  
Day(01/18/2020) = 7
```



Warning: .CurrentDay variable (System Enum) first value is Sunday(0). if you want to compare this variable with the result of the function do not forget to apply the offset.

Month(DATEVAR)

Extracts the [Month] value from a date variable

- Input: DATEVAR: date variable
- Output: Number corresponding to month number

Example:

```
DateVariable = 01/18/2020  
Month(DateVariable) = 1
```

Year(DATEVAR)

- Extracts the [Year] value from a Date variable
- Input: DATEVAR: date variable
- Output: Integer corresponding to year number

Example:

```
DateVariable = 01/18/2020  
Year(DateVariable) = 2020
```

LowerCase(STRINGVAR)

Converts the desired String to lower case characters

- Input: STRINGVAR: string variable, enum field variable
- Output: STRINGVAR in lower case

Example:

```
LowerCase("TesT stRING") = "test string"
```

UpperCase(STRINGVAR)

Converts the desired String to upper case characters

- Input: STRINGVAR: string variable, enum field variable
- Output: STRINGVAR in upper case

Example:

```
UpperCase("TesT stRING") = "TEST STRING"
```

Length(STRINGVAR)

Returns the count of character of the selected String variable

- Input: STRINGVAR: string variable, enum field variable
- Output: Integer corresponding to length of STRINGVAR

Example:

```
Length("TesT stRING") = 11
```

CharAt(STRINGVAR,INDEX)

Get a character of the selected string variable at a specified position

- Input: STRINGVAR: string variable, enum field variable
INDEX: number, integer variable, real variable rounded; INDEX range is zero-based (the first position is 0)
- Output: Single character string corresponding to character of STRINGVAR at INDEX position.
If INDEX is greater than the length of STRINGVAR, CharAt() returns an empty string.

Examples:

```
CharAt("TesT stRING",3) = "T"  
CharAt("TesT stRING",35) = ""
```

GetFieldOfEnum(ENUMVAR,INDEX)

Get field (string part) of the selected enum variable at specified index

- Input: ENUMVAR: enum variable.
INDEX: number, integer variable, real variable rounded; INDEX range is zero-based (the first enum value is 0)
- Output: String corresponding to field of ENUMVAR at INDEX.
If INDEX is outside of range, GetFieldOfEnum() returns an empty string.

Examples:

```
MyEnum = {"Medialon","Manager","v7"}  
GetFieldOfEnum(MyEnum,2) = "v7"  
GetFieldOfEnum(MyEnum,4) = ""
```

GetIndexOfEnum(ENUMVAR,STRING)

Searches for STRING in fields of the selected enum variable. If STRING is found in the variable, the index of the field is returned. If there are no matches, GetFieldOfEnum() returns -1.

- Input:
ENUMVAR: enum variable
STRING: string, string variable
- Output:
INDEX: Integer corresponding to the index of the ENUMVAR containing STRING. Enum index is zero-based (the first enum value is 0).



Special case: If several fields of ENUMVAR contain STRING, GetFieldOfEnum() returns the last field.

Examples:

```
MyEnum = {"Medialon","Manager","v7"}  
GetFieldOfEnum(MyEnum,"Manager") = 1  
GetFieldOfEnum(MyEnum,"manager") = -1  
GetFieldOfEnum(MyEnum,"test") = -1
```

Random(MAXNUMBER)

Returns a Random integer value between 0 and [current Value of selected variable]; the process is only available on Integers, Reals (units only) and Enums.

- Input:
MAXNUMBER: number, integer variable, real variable rounded
- Output:
Integer value

Example:

```
Random(123456) = 5637
```

Random(MAXUNIT,MAXDECIMAL)

Returns a Random real value between 0 and [current value of selected variables]; the process is only available on Integers, Reals (units only) and Enums.

- Input:
MAXUNIT: number, integer variable, real variable rounded. The part before decimal digit is

between 0 and (MAXUNIT-1).

MAXDECIMAL: number, integer variable, real variable rounded. The part after decimal digit is between 0 and (MAXDECIMAL-1).

- Output:
Real value

Examples:

`Random(123456, 34) = 5637.8`

`Random(1, 1000) = 0.387`

Sqrt(NUMBERVAR)

Returns square root of selected number or variable

- Input:
NUMBERVAR: Integer variable, real variable (must be greater than zero!)
- Output:
Real value

Example:

`Sqrt(123.456) = 11.11108`

Round(NUMBERVAR)

Rounds the selected number to the nearest integer. Round() uses the 'away from zero' method for tie- breaking (-1.5 rounds to -2).

- Input:
NUMBERVAR: Real variable
- Output:
Integer value

Examples:

`Round(1.6) = 2`

`Round(1.5) = 2`

`Round(1.4) = 1`

Ceil(NUMBERVAR)

Rounds the selected number to the superior integer

- Input:
NUMBERVAR: Real variable
- Output:
Integer value

Example:

$\text{Ceil}(1.6) = 2$
 $\text{Ceil}(1.4) = 2$
 $\text{Ceil}(1.0) = 1$

Floor(NUMBERVAR)

Rounds the selected number to the inferior integer

- Input:
NUMBER: Real variable
- Output:
Integer value

Example:

$\text{Floor}(1.6) = 1$
 $\text{Floor}(1.4) = 1$
 $\text{Floor}(1.0) = 1$

Abs(NUMBERVAR)

Returns the Absolute Value of selected number

- Input:
NUMBERVAR: Integer variable, Real variable
- Output:
Integer or Real value (depending on input type)

Examples:

$\text{Abs}(-12.3) = 12.3$
 $\text{Abs}(134) = 134$

Max(NUMBER1VAR,NUMBER2VAR)

Returns the greater value between 2 selected numbers

- Input:
NUMBER1VAR: Integer variable, Real variable
NUMBER2VAR: Integer variable, Real variable
- Output:
Integer or Real value (depending on input type)

Examples:

$\text{Max}(12.3, 4) = 12.3$
 $\text{Max}(-3, -134.34) = -3$

Min(NUMBER1VAR,NUMBER2VAR)

Returns the inferior value between 2 selected numbers

- Input:
NUMBER1VAR: Integer variable, Real variable
NUMBER2VAR: Integer variable, Real variable
- Output:
Integer or Real value (depending on input type)

Examples:

`Min(12.3,4) = 4`

`Min(-3,-134.34) = -134.34`

Sin(ANGLEVAR)

Returns the sine of selected angle; angle units in radians

- Input:
ANGLEVAR: Real variable
- Output:
Real value (Value min= -1, max= +1)

Examples:

`Sin(0) = 0.0`

`Sin(-1.25) = -0.94898`



Trigonometric functions (Sin, Asin, Cos, Acos, Tan, Atan) calculate angle units in radians. To convert degrees to radians, multiply by $\pi/180$. To convert radians to degrees, multiply by $180/\pi$.

Asin(REALVAR)

Returns the arc-sine of selected Real, result in radians

- Input:
REALVAR: Real variable (min= -1, max= +1)
- Output:
Real value

Examples:

`Asin(1) = 1.57080`

`Asin(-0.94898) = -1.24999`

Cos(ANGLEVAR)

Returns the cosine of the selected angle, angle units in radians

- Input:
ANGLEVAR: Real variable
- Output:
Real value (min= -1 max= +1)

Examples:

$\text{Cos}(1) = 0.54030$
 $\text{Cos}(-0.12) = 0.99281$

Acos(REALVAR)

Returns the arc-cosine of the selected Real, result in radians

- Input:
REALVAR: Real variable (min= -1 max= +1)
- Output:
Real value

Examples:

$\text{Acos}(1) = 0.00000$
 $\text{Acos}(-0.5) = 2.0944$

Tan(ANGLEVAR)

Returns the tangent of the selected angle, angle units in radians

- Input:
ANGLEVAR: Real variable (Values must be between $-\pi/2$ (-1.570796) and $\pi/2$ (1.570796), limits excluded)
- Output:
Real value

Examples:

$\text{Tan}(1) = 1.55741$
 $\text{Tan}(-0.12) = -0.12058$

Atan(REALVAR)

Returns the arc-tangent of selected Real, result in radians

- Input:
REALVAR: Real variable (Values must be between $-\pi/2$ (-1.570796) and $\pi/2$ (1.570796), limits excluded)
- Output:
Real value

Examples:

$\text{Atan}(1) = 0.78540$
 $\text{Atan}(0.14112) = 0.14019$

Ln(NUMBERVAR)

Returns the Napierian (natural) logarithm of the selected number

- Input:
NUMBERVAR: number, Integer variable, Real variable. Value must be greater than zero.
- Output:
Real value

Examples:

$\text{Ln}(1) = 0.0$
 $\text{Ln}(123.456) = 4.81588$

Log(NUMBERVAR)

Returns the logarithm (base 10) of the selected number

- Input:
NUMBERVAR: number, Integer variable, Real variable. Value must be greater than zero.
- Output:
Real value.

Examples:

$\text{Log}(1) = 0.0$
 $\text{Log}(12.5) = 1.09691$

Exp(NUMBERVAR)

Returns the Exponential function of the selected number.

- Input:
NUMBERVAR: number, Integer variable, Real variable
- Output:
Real value

Examples:

`Exp(0) = 1.0`
`Exp(12.34) = 228661.95206`

Char(ASCII CODE)

Returns the character of the decimal ASCII code.

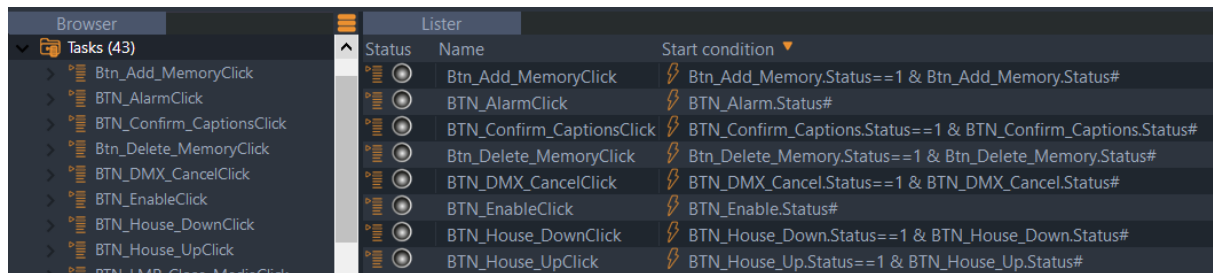
- Input:
ASCII CODE: number, Integer variable
- Output:
String value

Examples:

`Char(34) = "`
`Char(13) insert a carriage return (CR)`

6.3.5 Start Conditions

All tasks, whether step based or timeline, have a start condition that allows you to automatically execute the task. Start Conditions are written in the form of an Evaluation Expressions. When the evaluation returns the answer 'true' the task executes. The expression acts as a trigger.



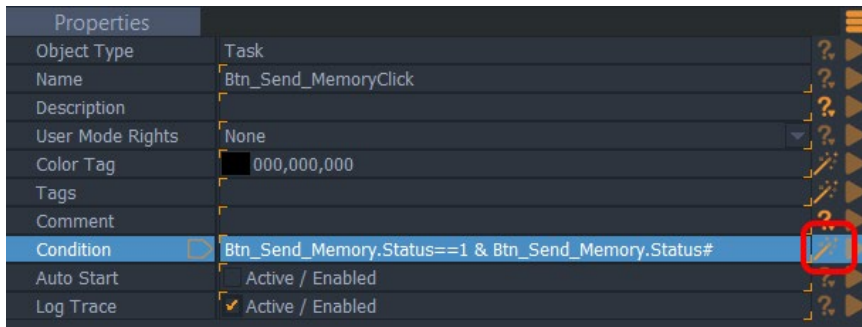
Browser	Tasks (43)	Lister	Status	Name	Start condition
	> Btn_Add_MemoryClick		⊙	Btn_Add_MemoryClick	Btn_Add_Memory.Status==1 & Btn_Add_Memory.Status#
	> BTN_AlarmClick		⊙	BTN_AlarmClick	BTN_Alarm.Status#
	> BTN_Confirm_CaptionsClick		⊙	BTN_Confirm_CaptionsClick	BTN_Confirm_Captions.Status==1 & BTN_Confirm_Captions.Status#
	> Btn_Delete_MemoryClick		⊙	Btn_Delete_MemoryClick	Btn_Delete_Memory.Status==1 & Btn_Delete_Memory.Status#
	> BTN_DMX_CancelClick		⊙	BTN_DMX_CancelClick	BTN_DMX_Cancel.Status==1 & BTN_DMX_Cancel.Status#
	> BTN_EnableClick		⊙	BTN_EnableClick	BTN_Enable.Status#
	> BTN_House_DownClick		⊙	BTN_House_DownClick	BTN_House_Down.Status==1 & BTN_House_Down.Status#
	> BTN_House_UpClick		⊙	BTN_House_UpClick	BTN_House_Up.Status==1 & BTN_House_Up.Status#
	> BTN_LMP_Close_MediaClick		⊙	BTN_LMP_Close_MediaClick	

Start Conditions can only function with Evaluation Expressions. If you use an Assignment Expression, the expression continually evaluates and the task is continually triggered.

Setting the Start Condition of a task

By default, there is no start condition for a task when it is created. In that case, the task can only be started manually via the Flash Controls or a Start Task command in another task. However, a task can be started automatically by specifying a Start Condition for the task.

To create a Start Condition, click on the 'magic wand' in the task properties for the condition property. This opens the 'Edit Expression' window for you to write out the evaluation expression.



A classic expression used to start a task is:

```
(Button_StartShow_1.Status==1) & (Button_StartShow_1.Status#)
```

This expression is divided into two parts that must evaluate true to make the entire expression true.

The first part compares the variable Button_StartShow_1.Status to 1 whereas the second part tests if Button_StartShow_1.Status has changed (since the last time the variable was evaluated).

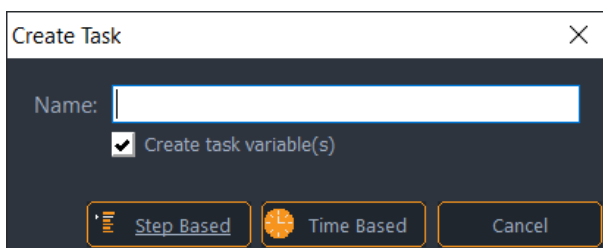
6.4 Step Based Tasks

Step based tasks are composed of a succession of command lines that execute consecutively. Step based tasks can be used as 'sub-programs' capable of performing simple to complex logical operations.

6.4.1 Creating a step based task

To create a step based task, the procedure is the same as the creation of time based tasks.

Select the 'Object/Task/New Task...' menu item or Select the Tasks item in the Object Browser and right-click to open the contextual menu. The Create Task dialog box opens:



Select the 'Step B'. type.

You can create variables that will reflect various properties of the task by checking the Create task variables box.

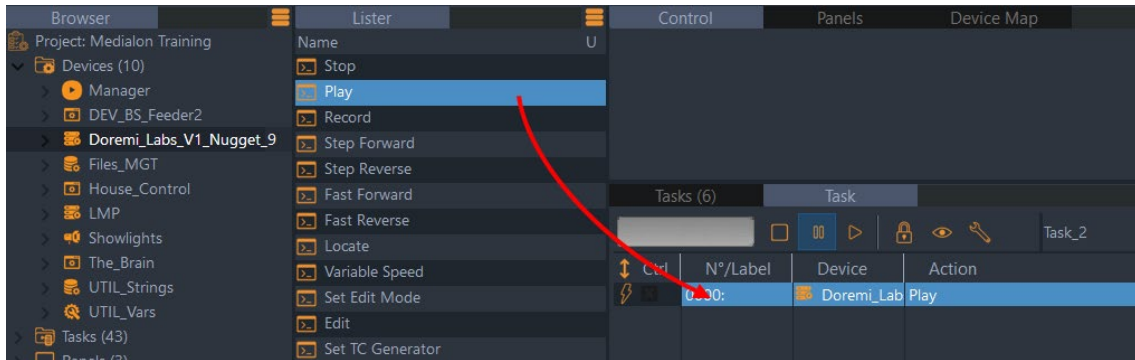
The step based tasks editor view is similar to the list view of time based tasks, but it has no time reference. The cues are played one after another in the order they are placed in the task.

Creating Cues

Creating cues in step based tasks is identical to creating them in time based tasks.

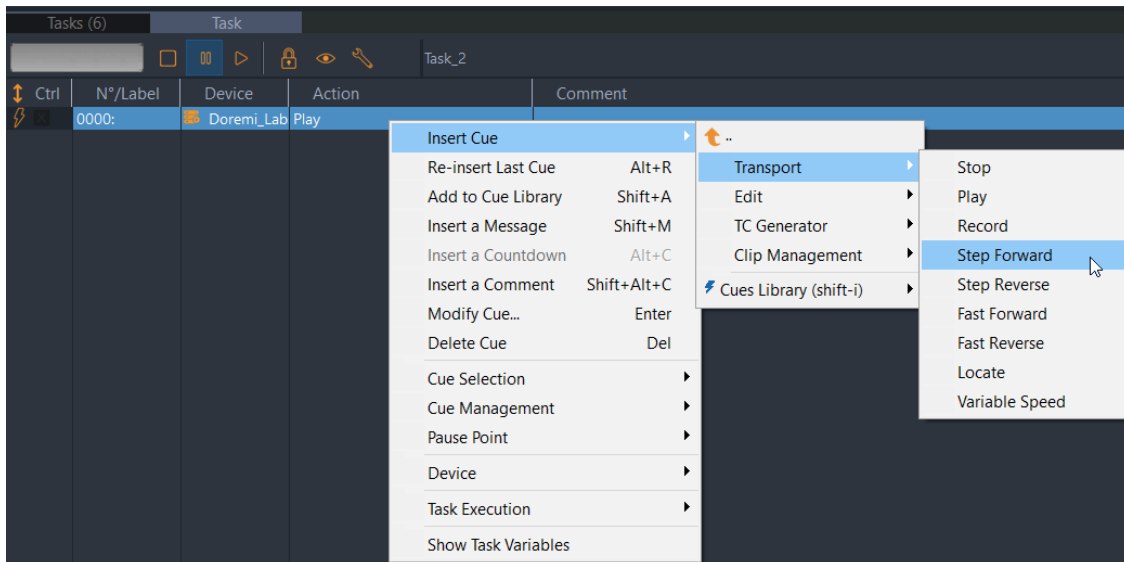
Method A

1. Select a command in the Lister tab.
2. 'Drag and Drop' the command where you want to insert the cue in the task.



Method B

1. Right-click inside the Task Editor tab and select 'Insert Cue'.
2. Browse to the desired device command and select it.



6.4.2 Selecting Cues

It is often necessary to select cues before using cue related commands; that is why the Task Editor provides a variety of ways to select cues. Selected cues are highlighted in orange or white, depending on your selected Theme.

To select cues:

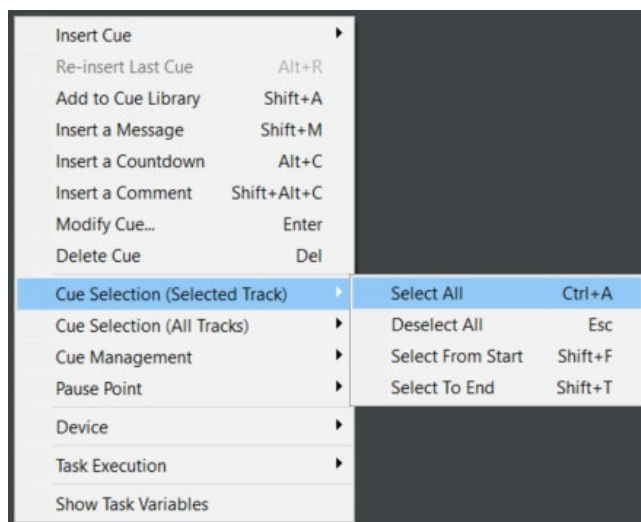
- Click on a cue to select it.
- Press the *Ctrl* key and click on another cue to add it to the selection.
- Press the *Shift* key and click on a cue to add all the cues between the previously selected cue and this cue.
- Type *Ctrl+A* to select all the cues in the task.
- Drag the mouse over the cues you want to select.

To deselect cues:

- Click on an empty part of the task editor to deselect all the cues.
- Press the *Ctrl* key and click on a selected cue to remove it from the selection.

The 'Cue Selection' menus

This contextual menu allows more sophisticated types of cue selection.



6.4.3 Managing Cues

Cut, Paste and Delete

Cut and paste commands are supported with familiar keyboard shortcuts:

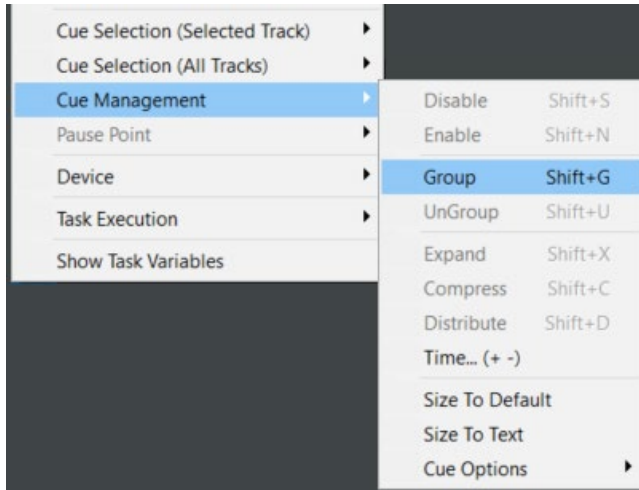
- Type the *Del* key to delete the selected cues.
- Type *Ctrl+C* to copy the selected cues.
- Type *Ctrl+X* to cut the selected cues.
- Type *Ctrl+V* to paste cues to the current position of the timeline or task.

Moving Cues

- Hold the *Alt* key down and use arrow keys to move the selected cues up or down.

The 'Cue Management' Menu

The Task Editor provides various commands in the 'Cue Management' submenu to automatically change relationships between cues.



Cue Disable/Enable

- *Disable / Enable* cues: Disabled cues are not executed during task execution.

To disable a selected cue:

- Type Shift+S.
- If Flash Control feature is activated, click on the Enable/Disable Flash Control Button
- Choose the 'Cues Management/Disable Cues' menu item.

To enable a selected cue:

- Type Shift+N.
- If Flash Control feature is activated, click on the Enable/Disable Flash Control Button.
- Choose the 'Cues Management/Enable Cues' menu item.

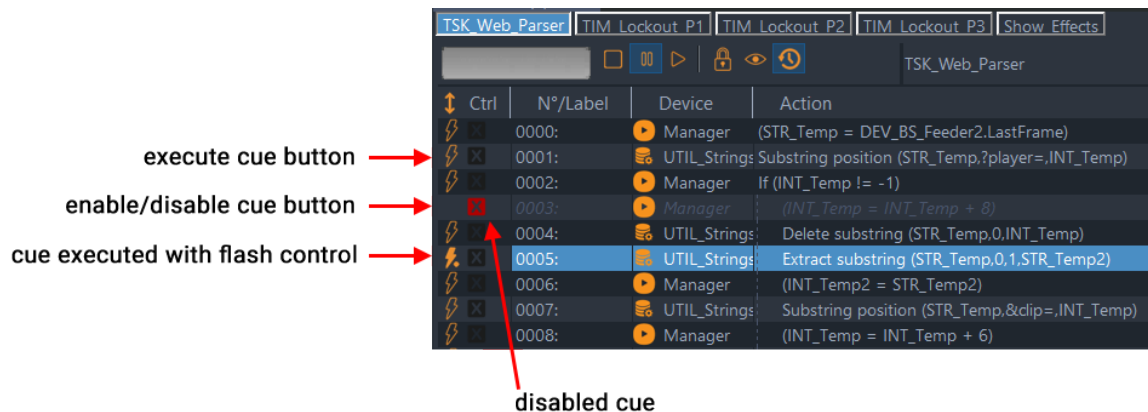
Grouping Cues

Cues may also be grouped. Once a group of cues is created, clicking on any cue of the group selects all the members of the group. This is particularly handy to make sure that the relationship between cues of a group is not modified by mistake while dragging cues.

- To group cues: select the cues to be grouped and either choose the Edit/ Group menu item or type *Ctrl+G*.
- To ungroup a group of cues: click on one cue of the group to select the whole group and either choose the Edit/Ungroup menu item or type *Ctrl+U*.


6.4.4 Flash Controls

The 'Flash Controls' give the capability of controlling cues in a task regardless of the current position or status.




The Flash Button on the left of each command cue allows you to execute the cue. The Round button allows you to disable (Red) or enable (Green) the cue. 'Message' cues do not have Flash Controls.

6.4.5 Blind Mode

Blind mode  allows you to edit and run your task without firing any cues.

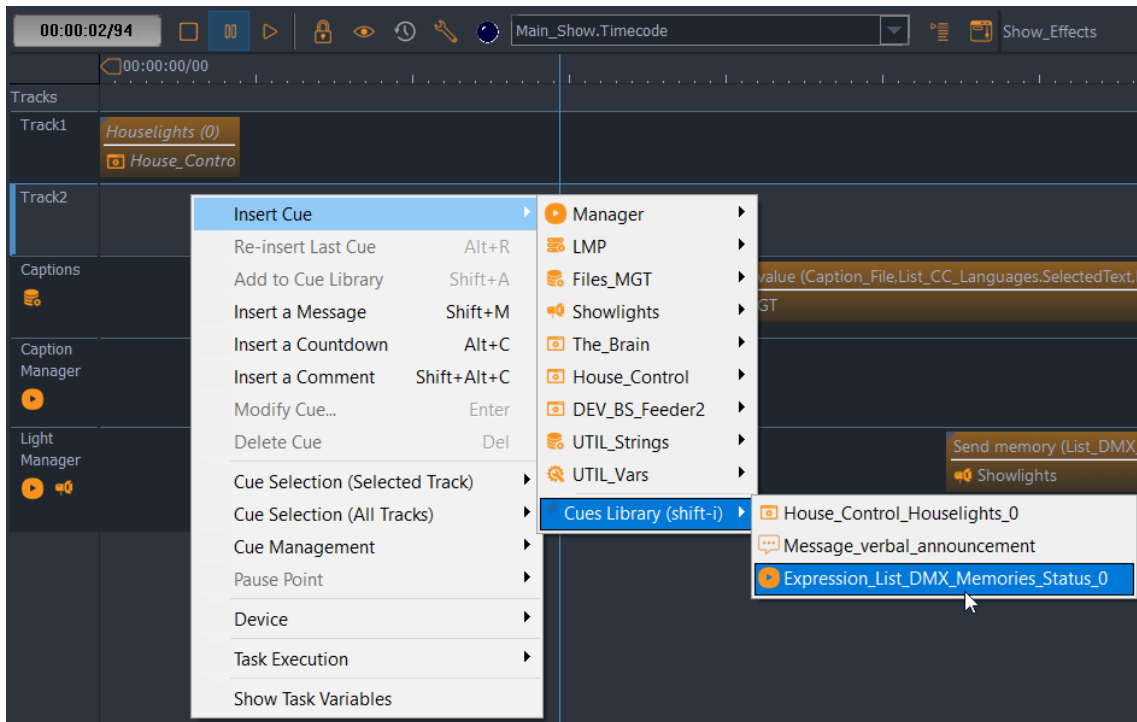
When blind mode is On, cues cannot be executed even with Flash commands. The blind mode affects the selected task, all other tasks in your project continue to be executed as normal, devices perform all commands received by the other tasks.

6.4.6 Lock Task

Lock Task  prohibits the task from launching automatically when the start condition becomes true or when a Start Task command is sent. The task may still be fired by using the play buttons in the Task Control Panel and Task Editor tab, or with the Flash Controls.

6.4.7 The Cue Library

The Cue Library is a convenient way to store cues that are often used. Cues can be easily added to and retrieved from the library by typing keyboard shortcuts: add a timeline cue to the library with *Ctrl+A*. See also [Cue Library](#).

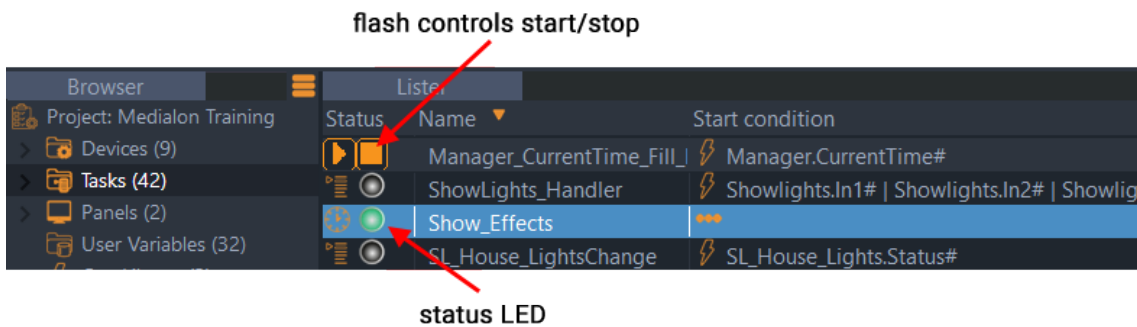


6.4.8 Pause Points

Pause Points allow you to debug complicated tasks. This function is not available in Run Mode. See also [Advanced Tools](#).

6.4.9 Tasks in a Project

All tasks created in a project are listed in the Browser Tab.



Select the Tasks node to display all tasks in the Lister.

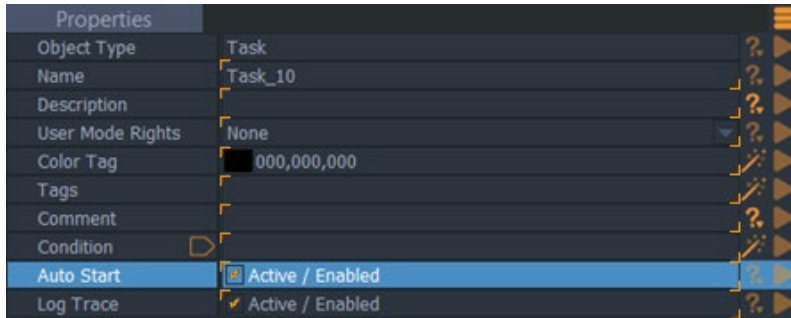
The status of each task is displayed by a LED (Dark Blue = Stopped, Yellow = Paused, Light Blue = Running)

Flash controls are available in the Lister tab to start or stop tasks.

Task Properties

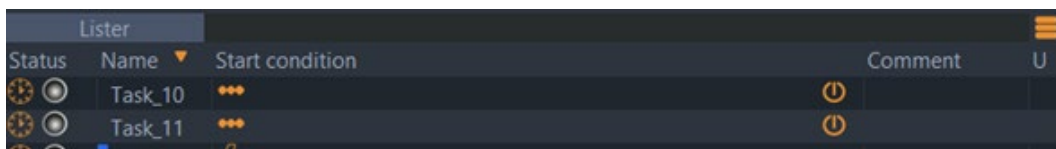
Expand the Tasks node in the Browser tab to see all tasks.

Select a task in the Browser or in the Lister; its properties are displayed in the Object Properties tab:



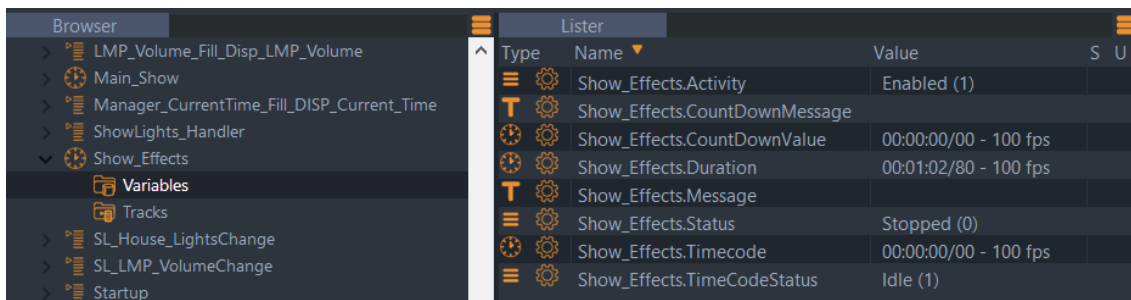
- **Name:** You can rename the task. Remember to validate your changes with the *Enter* key.
- **Description:** Description of the task
- **User Mode Rights:** Defines if the task is visible or editable in User Mode
- **Comment:** You can add a comment about the task
- **Condition:** Expression that triggers the task to execute when it becomes true
- **AutoStart:** Allows the task to automatically run when show enters Debug, User or Run modes without the need to set a start condition.

The Task Lister shows an icon indicating Auto Start  in the Start Condition column.



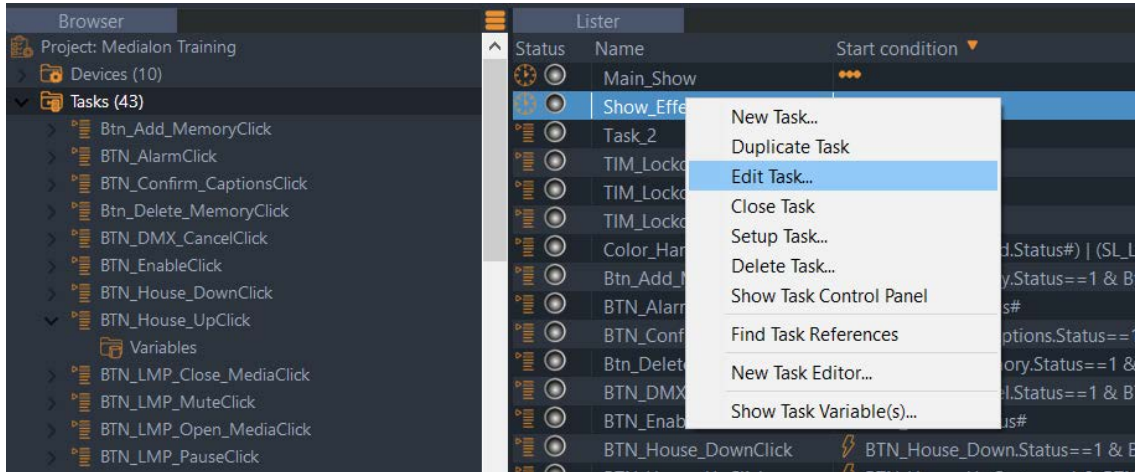
Task Variables

Select a task and expand it in the Browser tab to display the list of variables and the tracks (in the case of a time based task) in the Lister tab.



Task Management

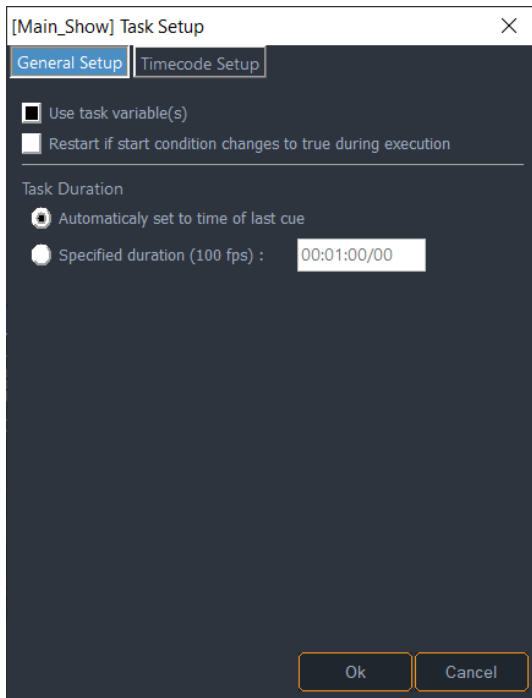
Once you have created your task, you may access the task management menu by right-clicking on the desired task in the Browser tab or in the Lister tab. The contextual menu appears, giving you access to the management options.



- *New Task*: Creates a new task
- *Duplicate Task*: Duplicates the selected task and names the duplicate with a numerical incremental value
- *Edit Task*: Opens the task editor in the Task tab or switches to the selected task's editor if it is already open
- *Setup Task*: Opens the selected task's setup window
- *Delete Task*: Deletes the selected task from the project after prompting for confirmation
- *New Task Editor*: Opens a new Task tab. With the new tab open, selecting Edit Task opens the Task Editor of the selected task in the new tab.
- *Show Task Variable(s)*: Displays variables of the selected task in the Lister.

6.4.10 Task Setup

Clicking on the Task Setup button or right-clicking on the task in the browser and selecting 'Setup Task...' opens a Task Setup dialog box for the currently selected task.



General Setup

Use Task Variables

If this option is checked, task variables are created.

Restart if start condition changes to true during execution

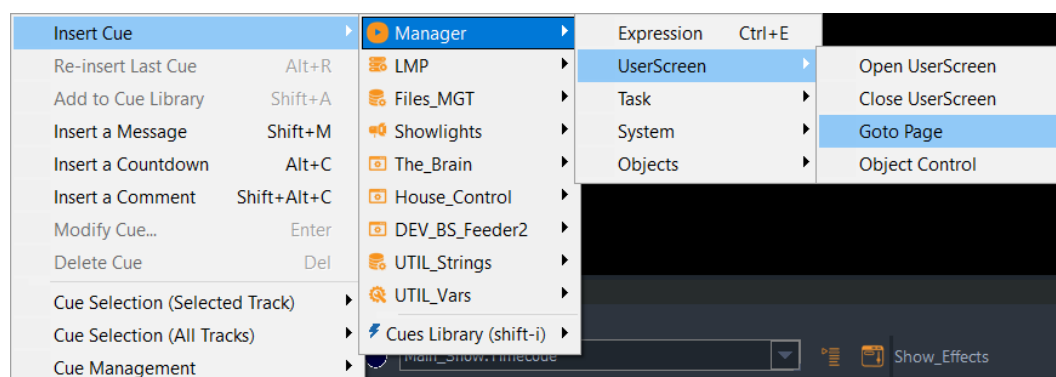
If this option is checked, a task is restarted when it reaches the end if its Start Condition changes to true while the task is running. This is useful to guarantee that a variable change which causes the condition to become true is correctly taken into account inside the task.

6.5 Device Manager/Showmaster

Medialon Manager V7 itself is viewed as a device and can receive commands as regular devices do. These commands control tasks, change device settings, and control user interfaces.

6.5.1 UserScreen Commands

This set of commands allows you to control the custom user interface.



Open UserScreen

From a user's point of view, the constituent parts of a program may be several 'UserScreens'. The appearance of these screens, linked to the user's requests (choice, various options), the requirements of the creator (organization of a game, of a dispatching, etc.), or practical necessities (information, warnings, etc.), is controlled by the 'Open UserScreen' command.

Parameters:

- *UserScreen Name*: [Type: String]: Name of UserScreen to be opened.
- *Position*: [Type: Enum]: Position mode, Available values:
 - 'Default': Uses default UserScreen position, does not force Left and Top parameters
 - 'Absolute': UserScreen position is forced by Left and Top parameters
- *Left*: [Type: Integer]: Left position of UserScreen
- *Top*: [Type: Integer]: Top position of UserScreen

There is no limit to the number of screens that can be opened simultaneously, except for the screen surface and thus the legibility of the displayed elements.

Close UserScreen

The opposite of the previous action; the closing of a 'UserScreen' is carried out by the 'Close UserScreen' command.

Parameters:

- *UserScreen Name*: [Type: String]: Name of UserScreen to be closed

Goto Page

You can display several 'UserScreens' simultaneously, but only one out of the 999 pages that a given 'UserScreen' can contain. The Goto Page command is used to select the desired page. This choice may be selected by the user or be provided by the programming.

Parameters:

- *UserScreen Name*: [Type: String]: Name of the UserScreen to go to page on
- *Page*: [Type: Integer]: Page number to go to

List Control

This command provides all the actions necessary for managing lists, for example: display, update, select/deselect, delete.

This command is only available in step based tasks.

Parameters:

- *List Name*: [Type: String]: Name of the list object in the UserScreen.
- *Action*: [Type: Enum]: Control to apply to the list, Available values:
 - 'Insert': Adds a new item to the list
 - 'Delete': Deletes an item from the list
 - 'Get Item': Retrieves the text content of a list item
 - 'Set Item': Replaces the text content of a list item
 - 'Get Count': Retrieves the total number of items in the list
 - 'Clear': Deletes all items in the list
 - 'Get Item List': Retrieves the text content of all list items
 - 'Set Item List': Replaces the text content of all list items
- *Index/Count*: [Type: Integer]: A bidirectional parameter, according to the action. It is either the index number for the item on which the action is to be executed or it is the result from the Get Count action.
- *Text*: [Type: String]: A bidirectional parameter, according to the action. It is either the text value you give to an item, or the result from the Get Item action.



Note: If you want to add a new item to the top of the list, the index parameter must be '0'. If you want to add a new item to the bottom of the list, the index parameter must be set to '-1'. This works only if the 'Sorted' property of the list object is not selected.

Object Control

This command can dynamically set or retrieve properties of dynamic graphic objects on UserScreens.

This command is available in step based tasks and in timelines.

Parameters:

- *Object Name*: [Type: String]: Name of the object in the UserScreen
- *Action*: [Type: Enum]: Action to perform, Available values:
 - 'Get': Returns the value of the property of the given object into the given variable
 - 'Set': Changes the property of the given object
- *Property*: [Type: Enum]: Type of text property, Available values:
 - 'Text'
 - 'Text Off'
 - 'Text On'
 - 'Visible': 0 is invisible, 1 is visible
 - 'Enable': enables/disables object interactivity. Object variables still work to update the objects, but mouse/touch actions do not affect the objects. The .click variable is not affected by the Enable property and will continue to work even if an object's Enable property is set to 0.
 - 'Color': string formatted as red, green, and blue integer values 0-255 separated by commas, for example: red is 255,0,0
 - 'Background Color': string formatted as red, green, and blue integer values 0-255 separated by commas, for example: red is 255,0,0
 - 'Color Off': string formatted as red, green, and blue integer values 0-255 separated by commas, for example: red is 255,0,0
 - 'Color On': string formatted as red, green, and blue integer values 0-255 separated by commas, for example: red is 255,0,0
 - 'URL'
 - 'Item Color Off': string formatted as red, green, and blue integer values 0-255 separated by commas, for example: red is 255,0,0
 - 'Item Color On': string formatted as red, green, and blue integer values 0-255 separated by commas, for example: red is 255,0,0
- *Value*: [Type: String]: A bidirectional parameter, according to the action. It is either the text, numeric, or RRR,GGG,BBB formatted color value you give to the object, or the result from the Get action.

The behavior of each property is slightly different per UserScreen object, and not all properties work with all objects.

Buttons:

- 'Text': Sets both On and Off text properties (gets only the 'Off' text property)
- 'Text Off': Sets/gets the Off text property

- 'Text On': Sets/gets the On text property
- 'Visible': Sets/gets the Visible property (0 is invisible, 1 is visible)
- 'Enable': Sets/gets the Enable property (0 is disabled, 1 is enabled)
- 'Text Color': Sets both On and Off text color properties (getting this property returns a null string)
- 'Text Off Color': Sets/gets the Off text color property
- 'Text On Color': Sets/gets the On text color property
- 'Color Off': Sets/gets the Off color property
- 'Color On': Sets/gets the On color property
- 'URL': Sets/gets the URL property for buttons (WebPanel buttons only)

Sliders:

- 'Visible': Sets/gets the Visible property (0 is invisible, 1 is visible)
- 'Enable': Sets/gets the Enable property (0 is disabled, 1 is enabled)
- 'Background Color': Sets/gets Background Color property

Digital Sliders:

- 'Visible': Sets/gets the Visible property (*0 is invisible, 1 is visible*)
- 'Enable': Sets/gets the Enable property (*0 is disabled, 1 is enabled*)
- 'Color': Sets/gets the 'on' color property
- 'Background Color': Sets/gets background color property

Lists:

- 'Text': Sets/gets the entire list of items
- 'Visible': Sets/gets the Visible property (*0 is invisible, 1 is visible*)
- 'Enable': Sets/gets the Enable property (*0 is disabled, 1 is enabled*)
- 'Text Off Color': Sets/gets the text color property of unselected items
- 'Text On Color': Sets/gets the On text color property of the selected item
- 'Color': Sets/gets the list's background color property
- 'Item Color Off': Sets/gets the unselected items' background color property
- 'Item Color On': Sets/gets the selected item's color property

Text Edit Boxes:

- 'Text': Sets/gets the text property (also available as a variable of the object)

- 'Visible': Sets/gets the Visible property (0 is invisible, 1 is visible)
- 'Enable': Sets/gets the Enable property (0 is disabled, 1 is enabled)
- 'Text Color': Sets/gets the text color property
- 'Background Color': Sets/gets background color property

Text Displays:

- 'Text': Sets/gets the text property (also available as a variable of the object)
- 'Visible': Sets/gets the Visible property (0 is invisible, 1 is visible)
- 'Text Color': Sets/gets the text color property
- 'Background Color': Sets/gets background color property

LEDs:

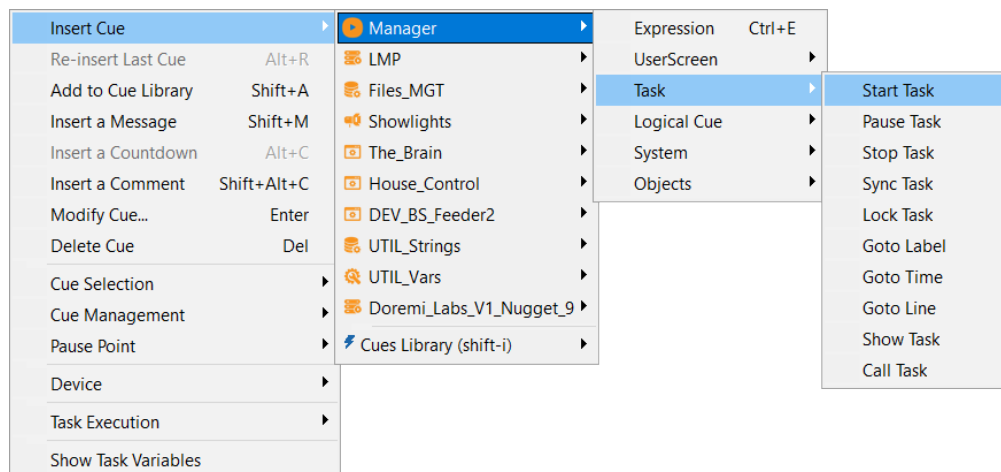
- 'Visible': Sets/gets the Visible property (0 is invisible, 1 is visible)
- 'Enable': Sets/gets the Enable property (0 is disabled, 1 is enabled)
- 'Color': Sets/gets the list of colors for all LED statuses (a line feed delimited list of red,green,blue formatted colors)

Gauges:

- 'Visible': Sets/gets the Visible property (0 is invisible, 1 is visible)
- 'Text Color': Sets/gets the text color property
- 'Color Off': Sets/gets the gauge's background color property
- 'Color On': Sets/gets the gauge's 'On' color property

6.5.2 Task Commands

This set of commands allows you to control the current task where the cues are inserted, but also gives you control of all the other tasks in your project.



Start Task

A task may be automatically started by the state of an object by using the Start Condition, which appears in the Task Lister Tab. But a task may launch another task using the 'Start Task' command. When programming the 'Start Task' command, you define the parameter of the task name to be launched. If the task is already running, this command has no effect.

Parameters:

- *Task Name*: [Type: String]: Name of task to be started

Pause Task

A task may be suspended momentarily and then continued again from where it was paused. To restart a task, use the 'Start Task' command from another task.

Parameters:

- *Task Name*: [Type: String]: Name of task to pause

Stop Task

This command not only stops a task, but it takes it back to the start (its line 0 or Time 00:00:00/00), so that the next 'Start Task' command makes it restart from the beginning.

Parameter:

- *Mode*: [Type: Enum]: Available values:
 - *Specified Task*: stops the task specified in the Task Name property
 - *All Tasks (Inclusive)*: Stops all tasks including the active task handling the Stop Task command
 - *All Tasks (Exclusive)*: Stops all tasks excluding the active task handling the Stop Task command
- *Task Name*: [Type: String]: Name of task to stop

Sync Task

Launches a 'time based task' by attributing a time-reference variable to it (reference Timecode). This time can be taken from any point on the 'timeline'. This command may be used when the clock of the 'timeline' concerned is already in execution.

Parameters:

- *Task Name*: [Type: String]: Name of task to synchronize
- *Sync Mode*: [Type: Enum]: Synchronization mode, Available values:

- *Internal*: The given task is internally synchronized
- *External*: The given task is synchronized to the variable 'Time Variable'
- *Time Variable*: [Type: Time]: Time variable used for synchronization reference

Lock Task

A task may be prohibited from starting using this command (Locked). Locked tasks cannot be started from Start Conditions or the Start Task command. This protection can be removed by using this command (Unlocked).

Parameters:

- *Task Name*: [Type: String]: Name of task to lock/unlock
- *Lock Mode*: [Type: Enum]: Available values:
 - *Unlocked*: The given task is unlocked
 - *Locked*: The given task is locked

Goto Label

A 'cue' is identified by a number (step based task) or a time (time based task). A 'Label' may be added to it. The advantage of a 'Label' associated with the 'cue' is that it cannot be changed accidentally as a 'cue' index can in the event of an insertion, or if the 'cue' is moved to a different time. The purpose of the 'Goto Label' command is to reach the cue associated with the Label and to launch the execution of the task from this Label. The same Label may be used in several different tasks.

Parameters:

- *Task Name*: [Type: String]: Name of task to go to label
- *Label*: [Type: String]: Label to go to in the task

Goto Time

This command is exclusive to the 'timeline' tasks and enables you to reach a given time in a time based task and launch its execution starting at this time.

Parameters:

- *Task Name*: [Type: String]: Name of task to go to time
- *Time*: [Type: Time]: Time to go to in the time based task

Goto Line

This enables you to access a 'cue' index in a step based task and launch the execution from that 'cue'.

To program this command, you must set the name of the task and the line number of the 'cue' in the task.

Parameters:

- *Task Name*: [Type: String]: Name of task to go to line number
- *Line*: [Type: Integer]: Line to go to in the step based task

Show Task

Opens or closes a Task Editor with the given task inside.

Parameters:

- *Task Name*: [Type: String]: Name of task to show/hide
- *Status*: [Type: Enum]: Show or hide, Available values:
 - *Show*: The Task Editor is opened
 - *Hide*: The Task Editor is closed
- *Left*: [Type: Integer]: Left position of Task Editor
- *Top*: [Type: Integer]: Top position of Task Editor
- *Width*: [Type: Integer]: Top position of Task Editor
- *Height*: [Type: Integer]: Top position of Task Editor



When the Task Name parameter is empty, the command is applied to the active task that executed the Show Task command.

Call Task

Causes the calling task to be suspended while the given task runs. This command is only available in step based tasks.

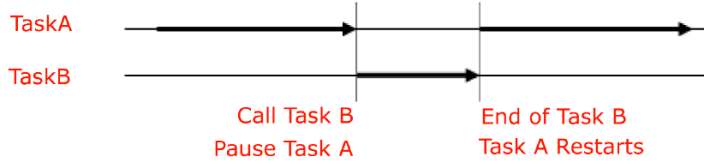
Parameters:

- *Task Name*: [Type: String]: Name of the task to call

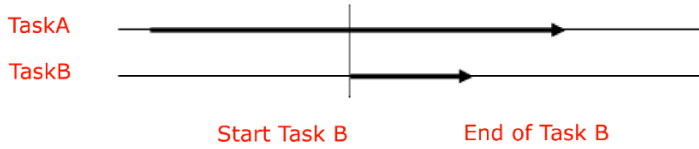
It is important to understand the difference between 'Call Task' and 'Start Task'. 'Call Task' is a command in a given task (A) that launches the execution of another given task (B), but it interrupts the running of task A while task B is in execution.

This is useful if you need to set a variable's value, and then call a task to process that variable before continuing with the original task. File, database, and network operations can sometimes take a few seconds to complete, so using Call Task ensures the original task (A) does not continue until the called task (B) has completed its work.

Call Task



Start Task



6.5.3 Expression

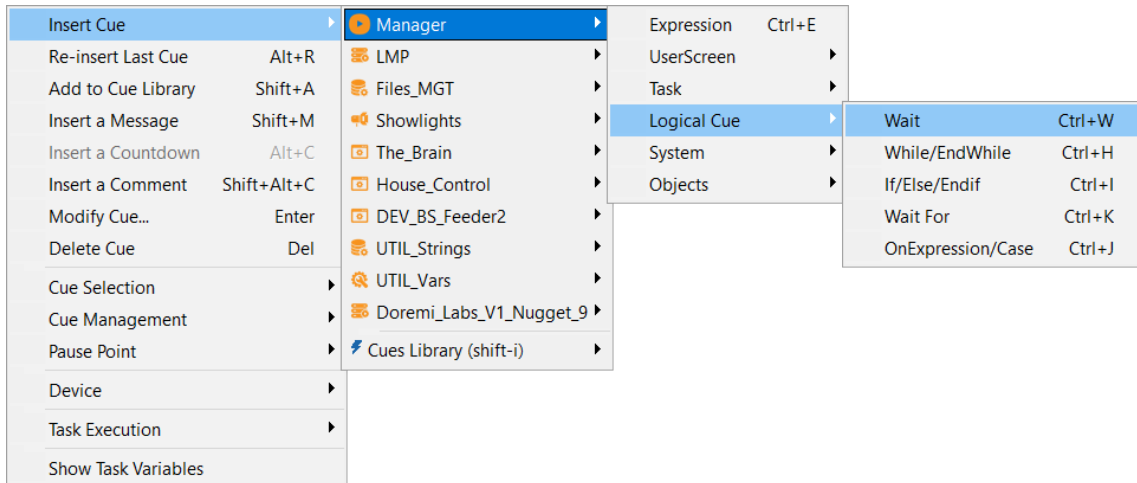
This provides the normal functions of computer programming, with Boolean operators. Writing expressions is accelerated by the provision of dialog boxes and drop-down menus, reducing the direct writing of code to a minimum.

Parameters:

- *Expression*: [Type: Private]: Click on the Magic wand button to open an Expression Edit window. See also [Expressions](#).

6.5.4 Logical Cues

This set of commands introduces conditional execution into your project.



These commands are only available in step based tasks.

Wait

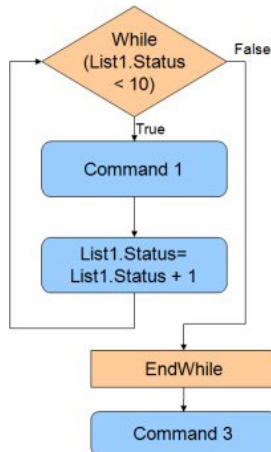
Enables you to introduce a delay time between two command lines, from 1/100 second to 99 hours, 59 minutes, 59 seconds and 99 hundredths of a second.

Parameters:

- **Time*: [Type: Time]: Amount of time the task should wait

While/EndWhile

This function enables you to create a loop: as long as the specified condition is true, the 'cues' between the 'While' cue and the 'EndWhile' cue execute. The loop stops when the 'While' condition is no longer true.

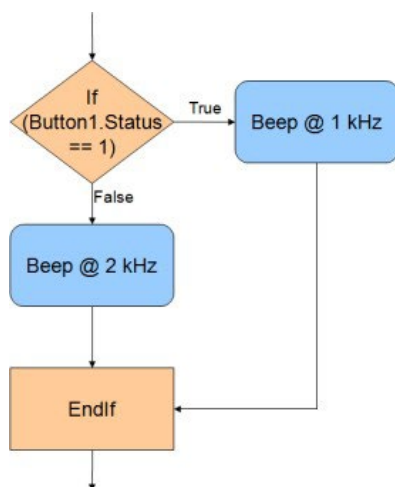


Parameters:

- *Condition*: [Type: Private]: A Comparison Expression to be evaluated for the While condition. Click on the Magic wand button to open an Expression Edit window (more details in the Expressions chapter).

If/Else/Endif

This cue triggers a command only if the required condition is true. Otherwise, another command may be activated. The 'Else' function is automatically created but can be deleted or ignored if it is not used.

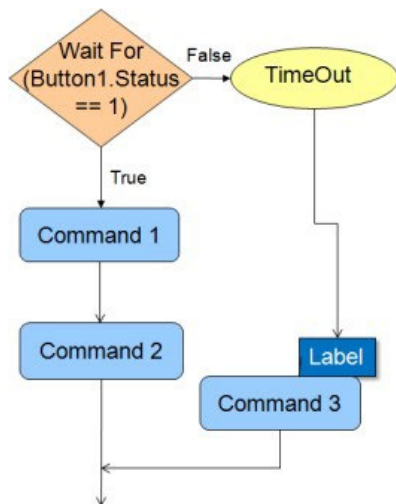


Parameters:

- *Condition*: [Type: Private]: A Comparison Expression to be evaluated for the If condition. Click on the Magic wand button to open an Expression Edit window (more details in the Expressions chapter).

Wait For

This function enables the task to wait for an expression (e.g. waiting for a device status) to be true, with a timeout and a label to go to in case of timeout.

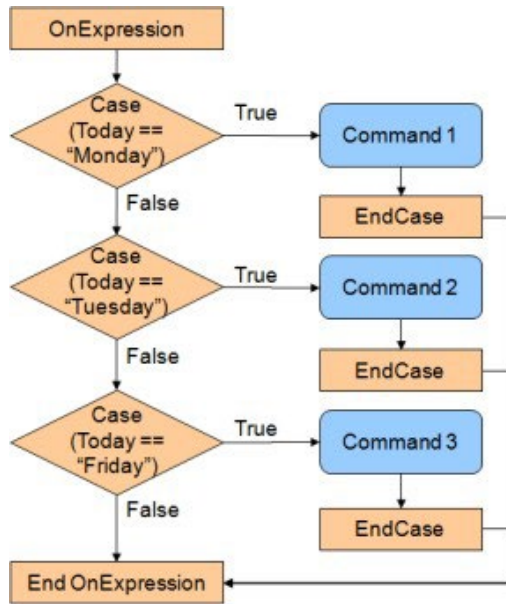


Parameters:

- *Condition*: [Type: Private]: A Comparison Expression to be evaluated as the Wait For condition. Click on the Magic wand button to open an Expression Edit window (more details in the Expressions chapter).
- *Timeout*: [Type: Time]: If the condition is not satisfied within this amount of time, the task jumps to the given label.
- *On timeout goto label*: [Type: String]: Label to go to when the timeout occurs. If no label is specified, the task executes the next cue in the task.

OnExpression/Case

This command block evaluates each 'Case' expression until it reaches an expression that is true or it reaches the end of block, 'EndOnExpression'.



Once a true case is reached, the cues inside the Case structure execute and upon reaching EndCase, execution jumps to the EndOnExpression cue.

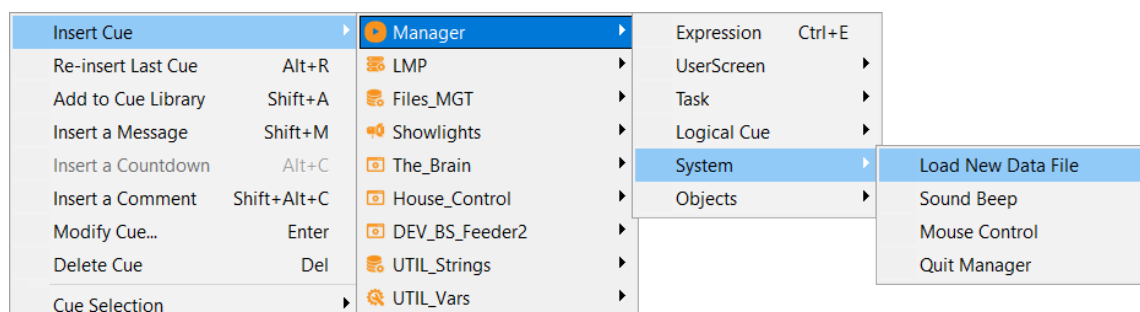
To ensure that at least one expression always executes in the event all other Cases fail, use Case(1) as the final Case in the OnExpression structure.

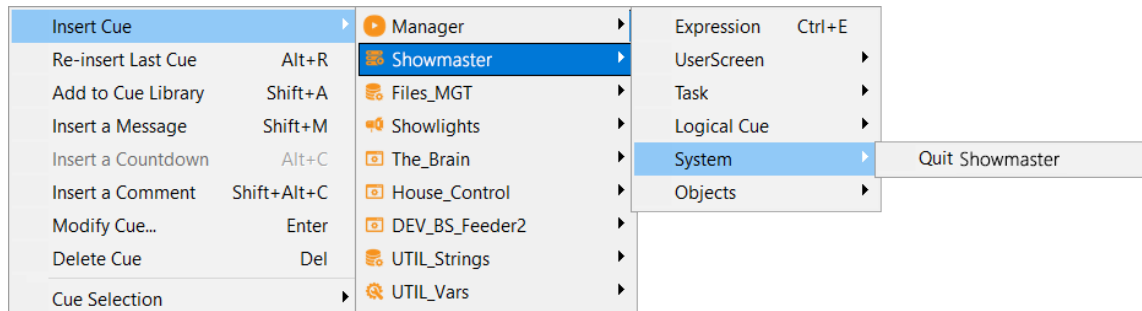
Parameters for each Case statement:

- *Condition*: [Type: Private]: A Comparison Expression to be evaluated in each Case. Click on Magic wand button to open Expression Edit window (more details in the Expressions chapter).

6.5.5 System Commands

This set of Manager commands act on the host machine.





Sound Beep

This command allows you to send a sound beep directly from the Motherboard speaker with the desired parameters (frequency and duration) in increments of 1/100 second.

Parameters:

- *Frequency*: [Type: Integer]: Sound frequency from 20Hz to 20KHz
- *Duration*: [Type: Integer]: Duration of sound in 1/100s
- *Wait*: [Type: Enum]: This parameter has no effect in time based task, Available values:
 - *Yes*: Current task is paused during sound duration
 - *No*: Current task is not paused during sound duration

Quit Manager

This command allows you not only to quit Medialon Manager and also quit Windows, but to also Power Off or Reboot the host computer.

This command is only available in step based tasks in Manager devices.

Parameters:

- *Shutdown PC*: [Type: Enum]: Shutdown mode, Available values:
 - *No*: Only Manager quits
 - *Shutdown*: Windows is asked to shutdown
 - *Reboot*: Windows is asked to reboot
 - *Power Off*: Windows is asked to shutdown and power off the PC
 - *Log Off*: Windows is asked to log out of the Windows session

This command is only executed in RUN mode.

Quit Showmaster

This command allows you to Shutdown or Reboot Showmaster. This command is only available in StepBased tasks in Showmaster devices.

Parameters:

- *Mode*: [Type: Enum]: Quit mode, Available values:
 - *Shutdown*: Showmaster is asked to shutdown
 - *Reboot*: Showmaster is asked to reboot



This command is only executed when Showmaster is disconnected from Showmaster Editor.



Mouse Control

This command changes the mouse cursor and mouse position on the Local Panel. This command is only available in Medialon Manager step based tasks.

Parameters:






- *UserScreen Name*: [Type: String]: UserScreen Name that serves as an X,Y reference by the command. If this property is empty, the entire Local Panel area is referenced.



Only Local Panel's UserScreen can be selected, this command does not work in Web Panels.

- *Cursor*: [Type: Enum]: Type of cursor to set, Available values:
 - *Default*: Default Windows cursor
 - *None*: Cursor is invisible

	Arrow
	Cross
	I Beam
	Size NESW: Size arrow, Northeast-Southwest
	Size NS: Size arrow, North-South
	Size NWSE: Size arrow, Northwest-Southeast
	Size WE: Size arrow, West-East
	Up Arrow
	Hour Glass: Wait cursor
	Drag
	No Drop
	H Split
	V Split
	Multi Drag
	SQL Wait

-  No: No cursor
-  App Start
-  Help
-  Hand Point
-  Size All

- *User (File)*: The cursor is in a bitmap file; the file path is indicated in field Cursor File
- *X Position*: [Type: Integer]: Horizontal position for the cursor. -1 means no change for this position.
- *Y Position*: [Type: Integer]: Vertical position for the cursor. -1 means no change for this position.



X and Y Positions are relative to the top left corner of the indicated UserScreen. If no UserScreen is specified, X and Y Positions are relative to the entire Local Panel area.

Load New Data File

When persistent variables are created, a file that saves their values is created (.mngVarStor).

The 'Load New Data File' command enables you to change the targeted storage file, updating the variables concerned, adapting them to a user or reopening a query aborted some time before.

This command can be used to create user profiles, for example, according to certain criteria (centers of interest, access rights, etc.).

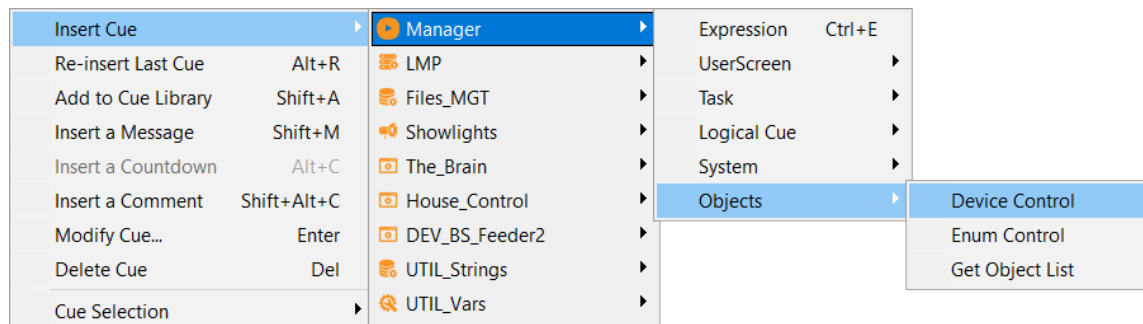
This command is only available in Medialon Manager step based tasks.

Parameters:

- *File Name*: [Type: String]: Path name of Persistent variable file

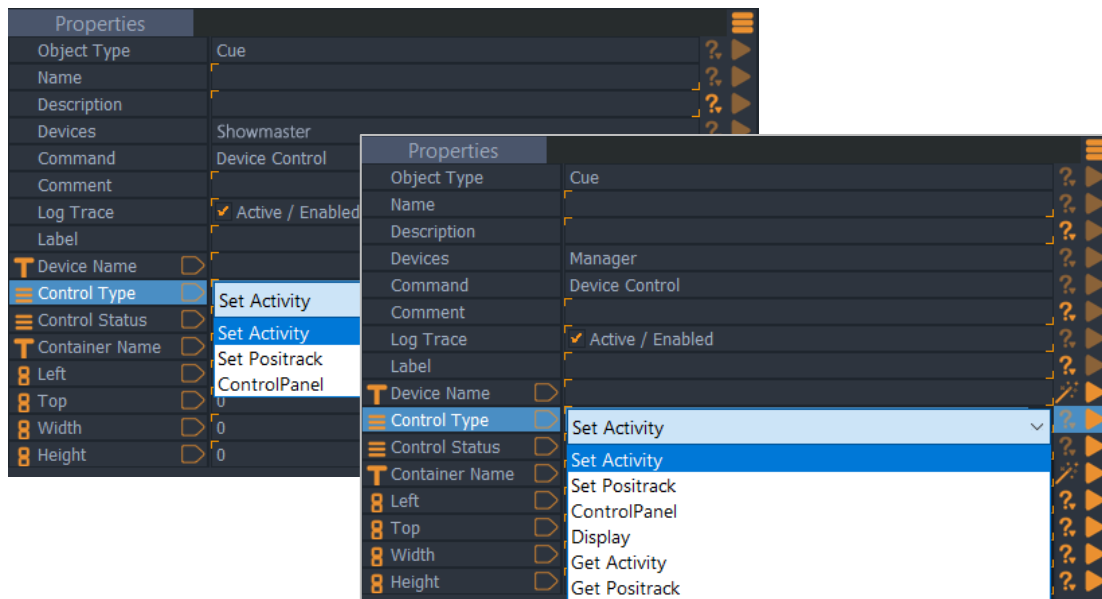
6.5.6 Object Commands

Medialon Manager can take control of devices and objects.



Device Control

This command is used to change the behavior of or to control a device. This command changes a Device's behavior and can open Control Panels.



If Device Activity is disabled, no commands generated by Medialon Manager V7 are sent to the real world device.


Positrack can be disabled for a specific device so that no repositioning commands are sent when a jump is made in a timeline.



Device Control Panels can be opened or set to a container in a LocalPanel UserScreen.

Parameters:

- *Device Name*: [Type: String]: Name of device to control
- *Control Type*: [Type: Enum]: Type of control to apply to the device, Available values:
 - *Set Activity*: enables or disables device activity
 - *Set Positrack*: enables or disables the device positrack mechanism
 - *Control Panel*: shows or hides the device Control Panel
 - *Display*: shows or hides the device display (for example, the video window of the Local Media Player MxM)
 - *Get Activity*: gives device activity status
 - *Get Positrack*: gives device positrack status
- *Control Status*: [Type: Enum]: Status of select control, Available values:
 - *Enable*: The control type is enabled
 - *Disable*: The control type is disable

-  *Container Name*: [Type: String]: Name of container in the UserScreen into which the Control Panel will be opened.
- *Left*: [Type: Integer]: Left position of Control Panel if no Container is specified
- *Top*: [Type: Integer]: Top position of Control Panel if no Container is specified
- *Width*: [Type: Integer]: Width of Control Panel if no Container is specified and the Control Panel is re sizable
- *Height*: [Type: Integer]: Height of Control Panel if no Container is specified and the Control Panel is re sizable

Enum Control

This command provides all the actions necessary to manage an Enum variable, for example: display, update, select/deselect, delete.

Parameters:

- *Enum Name*: [Type: String]: Name of Enum variable
- *Action*: [Type: Enum]: Control to apply the Enum, Available values:
 - *Insert*: Add a new value to the enum
 - *Delete*: Delete a value from the enum
 - *Get Enum*: Retrieve the text content of a given value
 - *Set Enum*: Replace the text content of a given value
 - *Get Count*: Retrieve the count of values in the enum
 - *Clear*: Delete all values from the enum
 - *Get Enum List*: Retrieve the text content of all values
 - *Set Enum List*: Replace the text content of all values
- *Index/Count*: [Type: Integer]: A bidirectional parameter, according to the action. It is either the Index value on which the action is made or the result from the Get Count action
- *Text*: [Type: String]: A bidirectional parameter, according to the action. It is either the text you give to the value, or the result from the Get Enum action



If you want to add a new item to the top of the enum, the index parameter must be '0'. If you want to add a new item to the bottom of the list, the index parameter must be set to '-1'.

Get Object List

Retrieves the list of objects available in your project. Parameters:

- *Parent Object Name*: [Type: String]: The starting point for establishing the Object List. If empty, the search is performed on the whole project. Use the wizard to select a parent object.

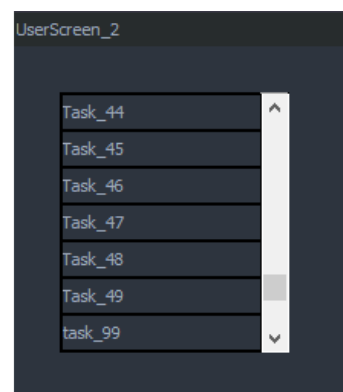
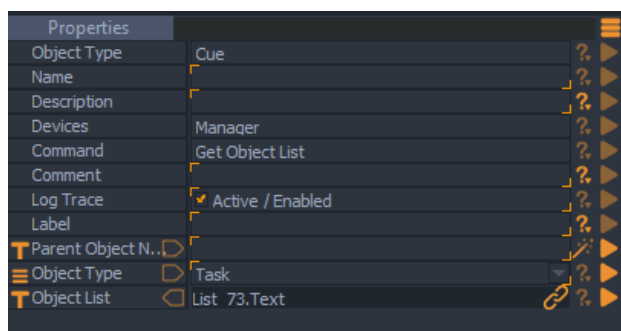
- *Object Type*: [Type: Enum]: Type of Object to be listed, Available values:
 - *All Objects*: No filter is applied
 - *Device*: Get the list of devices contained in the Parent Object
 - *Task*: Get the list of tasks contained in the Parent Object
 - *Variable*: Get the list of variables contained in the Parent Object
 - *User Variable*: Get the list of user variables contained in the Parent Object
 - *Time based task*: Get the list of time based tasks contained in the Parent Object
 - *Stepbased task*: Get the list of step based tasks contained in the Parent Object
 - *UserScreen*: Get the list of UserScreens contained in the Parent Object
 - *Button*: Get the list of buttons contained in the Parent Object
 - *Slider*: Get the list of sliders contained in the Parent Object
 - *Digital Slider*: Get the list of digital sliders contained in the Parent Object
 - *Display*: Get the list of displays contained in the Parent Object
 - *LED*: Get the list of LEDs contained in the Parent Object
 - *Gauge*: Get the list of gauges contained in the Parent Object
 - *Container*: Get the list of containers contained in the Parent Object
 - *List*: Get the list of lists contained in the Parent Object
 - *Edit*: Get the list of text edit objects contained in the Parent Object
 - *Any Graphic Object*: Get the list of all graphic objects contained in the Parent Object
- *Object List*: [Type: String]: The result of the command

Parent objects and allowed object types

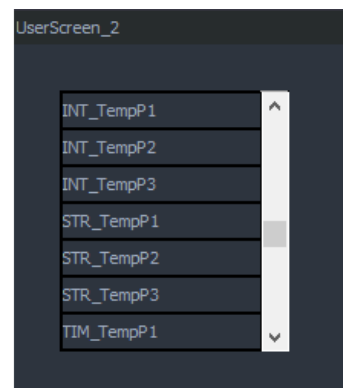
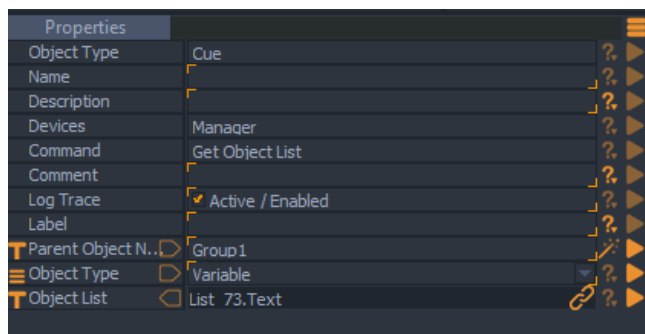
Parent Object	Device	Task	Variable	UserScreen	Graphical Object
Project	X	X	X	X	X
Device			X		
Task			X		
UserScreen			X		X
Graphical Object			X		
User Group		X	X	X	

Examples:

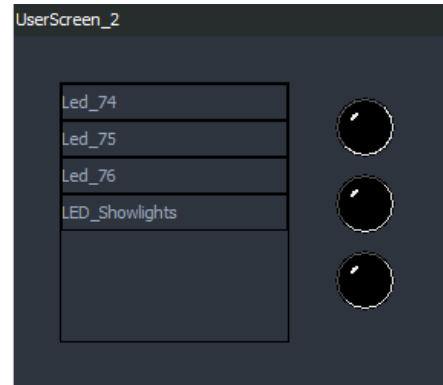
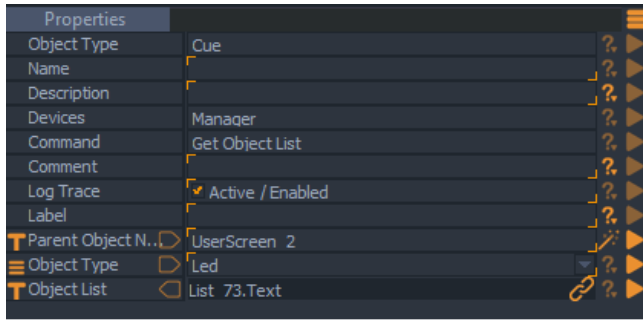
Get list of tasks in your project:



Get list of variables in a User group:



Get list of LEDs in a UserScreen:



6.5.7 Manager Variables

Manager V7 returns information via some variables.

Lister				
Type	Name	Value	S	U
	Manager.CurrentTime	10:29:57/73 - 100 fps		
	Manager.CurrentDay	Monday (1)		
	Manager.CurrentDate	23/05/2022 - Monday (1)		
	Manager.CurrentStatus	Stop (0)		
	Manager.CurrentProjectFile	C:\Users\Public\Documents		
	Manager.CurrentProjectTitle	Training		
	Manager.ConnectedPanels	0		
	Manager.MemoryUsage	47%		
	Manager.CPUUsage	4		
	Manager.Hostname	LaptopPC5		
	Manager.HostIPAddresses	10.100.101.41		
	Manager.ConnectedWebpanelsIP			



Screen, mouse, and keyboard Manager variables in previous version of Medialon Manager are now LocalPanel.variables. See also [Panels Variables](#).

Status

This variable indicates in which mode your project is. This variable is used to define the start condition of an initial task in a project.

- Name: Manager.CurrentStatus
- Type: Enum, Available Values:
 - Stop (0)
 - Debug (1)
 - Run (2)
 - User Mode (3)

Date and Time

These variables return date and time of Medialon Manager host or Showmaster V2.

Current Date

- *Name:* Manager.CurrentDate
- *Type:* Date

Current Time

- *Name:* Manager.CurrentTime
- *Type:* Time

Current Day

- *Name:* Manager.CurrentDay
- *Type:* Enum, Available Values:
 - Sunday (0)
 - Monday (1)
 - Tuesday (2)
 - Wednesday (3)
 - Thursday (4)
 - Friday (5)
 - Saturday (6)

Current Project

Project Title

Title of the loaded project, this title can be changed in the Project Preferences. See also [Preferences](#).

- *Name:* Manager.CurrentProjectTitle
- *Type:* String

Project File

Path and file name of the loaded project.

- *Name:* Manager.CurrentProjectFile
- *Type:* String

Connected Panels

This variable returns how many panels are connected.

Automation Programming

- *Name:* Manager.ConnectedPanels
- *Type:* Integer

This variable is the sum of all connection with WebPanel, Medialon Manager Panel and OpenCAP.

T	⚙️	Manager.CurrentProjectFile	C:\Users\Public\Document
T	⚙️	Manager.CurrentProjectTitle	Training
B	⚙️	Manager.ConnectedPanels	7
T	⚙️	Manager.MemoryUsage	47%

Connected Panel(s)	1
Connected WebPanels(s)	6
Connected OpenCAP(s)	0
Granted Panels/OpenCAP(s)	10

The number of connections is limited.

- 2 connections are allowed by default with Medialon Manager V7 LITE
- 10 connections are allowed by default with Medialon Manager V7 PRO.



Granted connections for Medialon Manager Lite and Pro can be increased in batches of 10 connections, contact your Medialon dealer for more information.

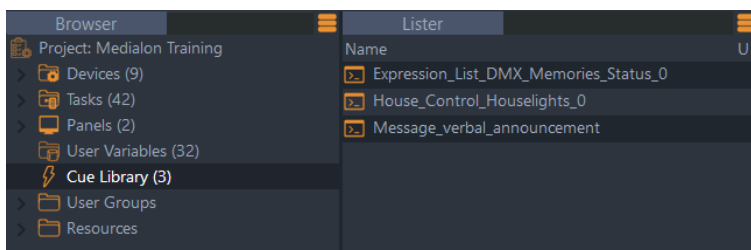
7 Advanced Programming

This section will teach you how to get the most of your Medialon Manager software using programming wizards, network interconnection, importing and exporting of sub programs and more.

7.1 Cue Library

7.1.1 The Cue Library

During the programming process, you will find that some cues are used several times within the same project. Medialon Manager V7 offers a Cue Library where commonly used cues can be stored.



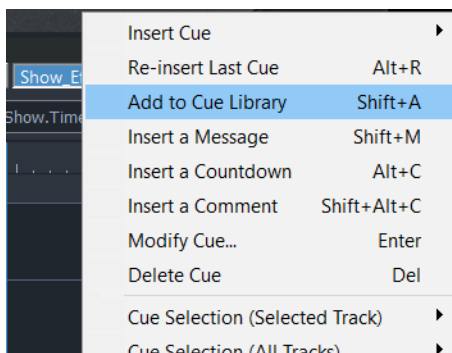
Cues are easily added to the library and retrieved by typing keyboard shortcuts or through drag and drop gestures.

The Cue Library is saved in the project file so that its contents remain when the project is reloaded.

7.1.2 Adding Cues to the Cue Library

From a Task

Select a cue and type *Shift+A*. The cue is added to the cue library. Alternatively, the 'Cue Library/ Add Selected Cues' menu item from the cue contextual menu can be used.





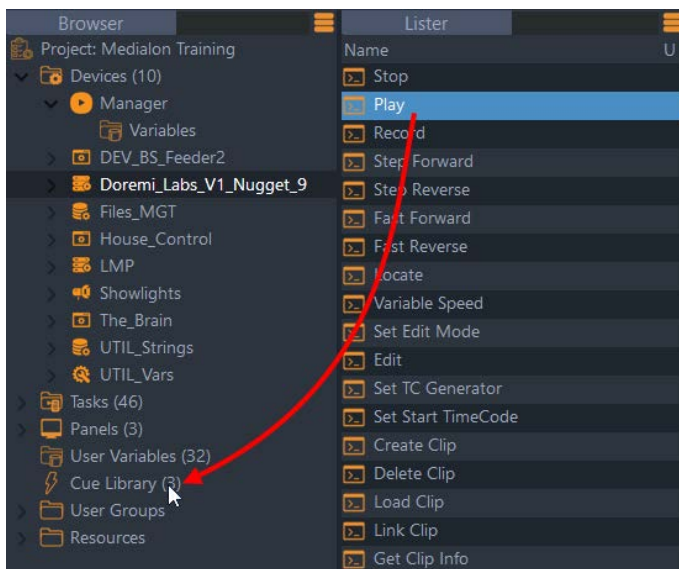
Structured Logical Cues (If/Else/EndIf, While/EndWhile and OnExpression...) cannot be added to the Cue Library.

From The Device Command List

You can add Device Commands directly to the Cue Library to prepare your library before task creation.

You can, for instance, prepare a lighting memory with a DMX Send Cue command and 'store' it in the Cue Library.

1. Select a Device in the Object Browser
2. Select a command in the Object Lister
3. Drag and Drop it into the Cue Library

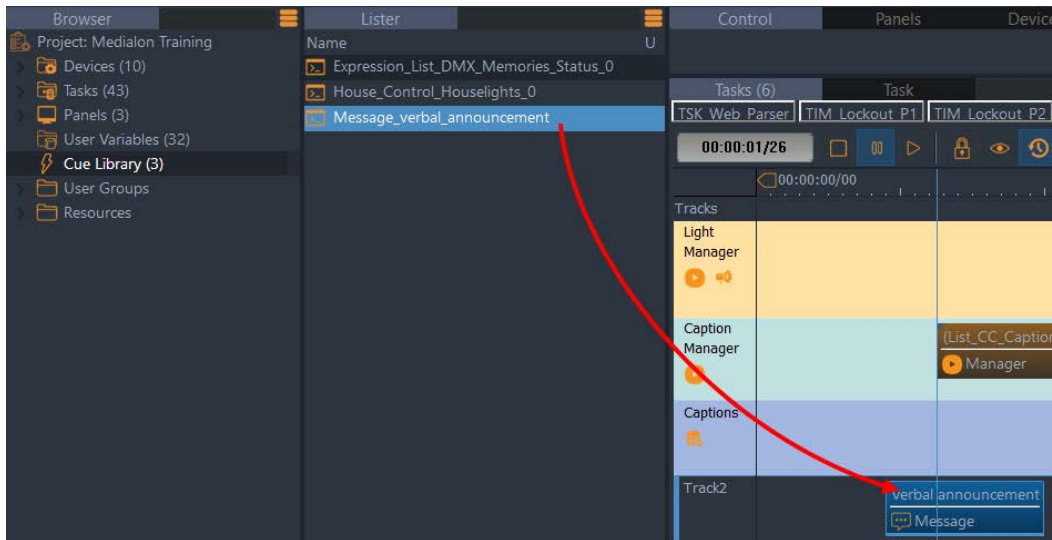


7.1.3 Inserting Cues from the Cue Library

There are two methods for adding cues from the Cue Library to your tasks.

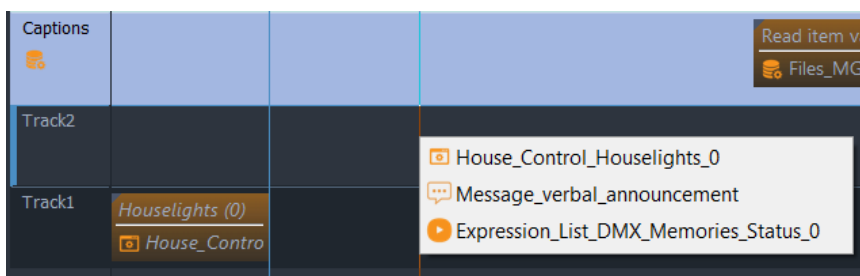
Method A

1. Select the Cue Library in the Browser tab to display the cues in the Lister tab
2. Select a cue in the Lister tab
3. Drag and Drop the cue into an open task in the Task Editor

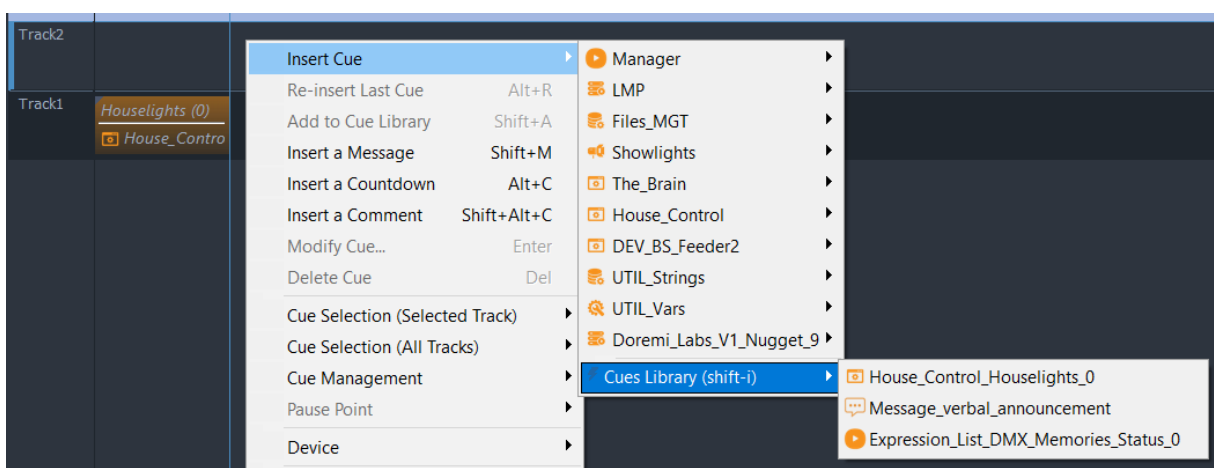


Method B

While editing a task, type the keyboard shortcut *Shift+I* to display the Cue Library window and then click on the cue you want to insert or navigate the list with the arrow keys and select by pressing Enter; the cue is added to the task.



Alternatively, the 'Insert Cue / Cue Library' menu item from the cue context menu can be used.



7.1.4 Testing Cues from Library

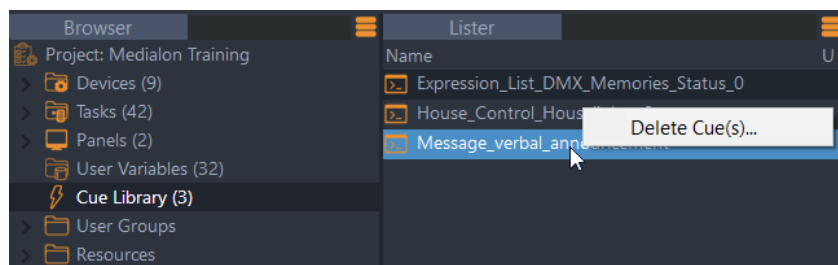
Medialon Manager allows the user to test cues from the Library.

1. Select a cue from the Library
2. Edit parameters, if needed
3. Click the *Test command* button in Object Properties tab



7.1.5 Deleting Cues from Library

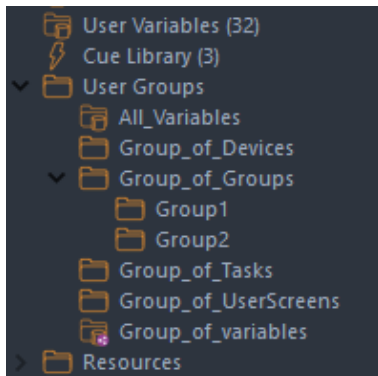
To delete a cue from the Cue Library, select it and use the right-click menu:



Note that multi-selection is available if you need to delete several cues.

7.2 User Groups

You may want to set up the structure of your project in separate parts, like show timelines, preshow devices, data handling processes, etc. With User Groups, you can create groups and sub-groups of objects of the same type (devices, variables, tasks or UserScreens). Grouping allows you to organize information that is logical for your project. When objects are added to User Groups, they are not moved, rather a shortcut to the object is placed in the User Group. This allows objects to have shortcuts in multiple User Groups.

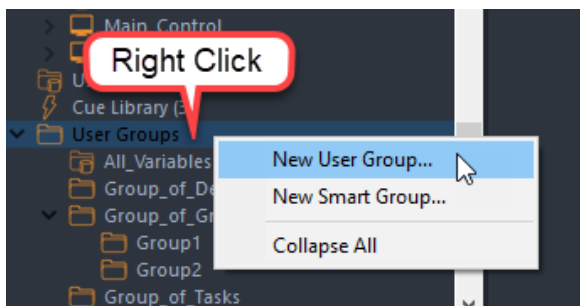


By default, Medialon Manager V7 creates one User Group containing all variables in the project.

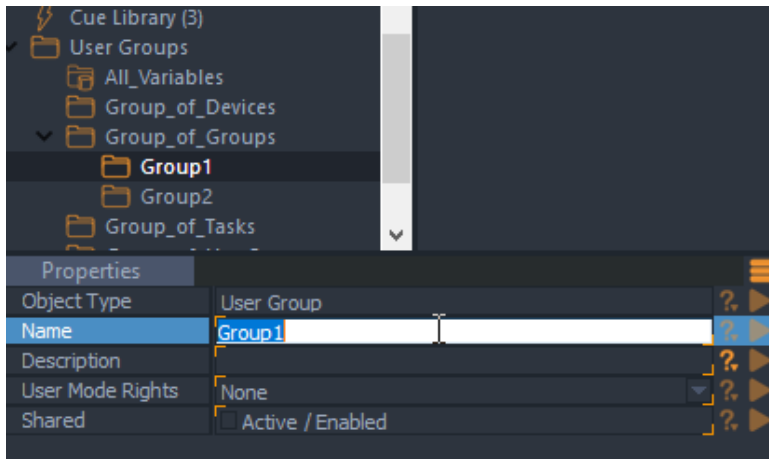
User Groups are also used to share objects with other Medialon devices, like Showmaster, Medialon Manager Lite, and Medialon Manager Panel. It is also the way information can be easily shared to third party software like Barco Overture.

7.2.1 Creating User Groups

Select the User Groups node and right-click to create a new User Group.



The name of the new group can be modified in the Properties tab.

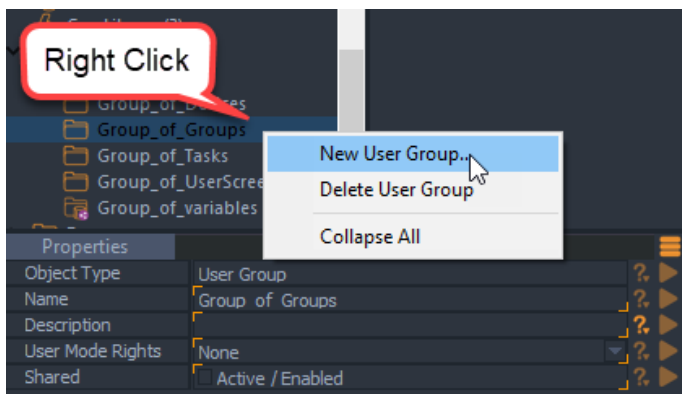


User Mode Rights defines if this User group is visible in User Mode. Note that the contents of the User group must also have their User Mode Rights set to View or they will not be visible even though the User group is.

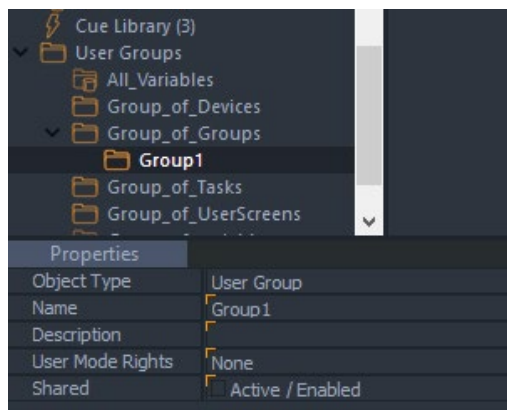
Sub-Group Creation

User Groups can only contain one type of object. If you need to group several types of objects, you can create sub-groups, each containing one type of object.

1. Select or create an empty User Group in the Browser
2. Right-click to create a new group



3. Rename the new sub-group



7.2.2 Adding Objects to User Groups

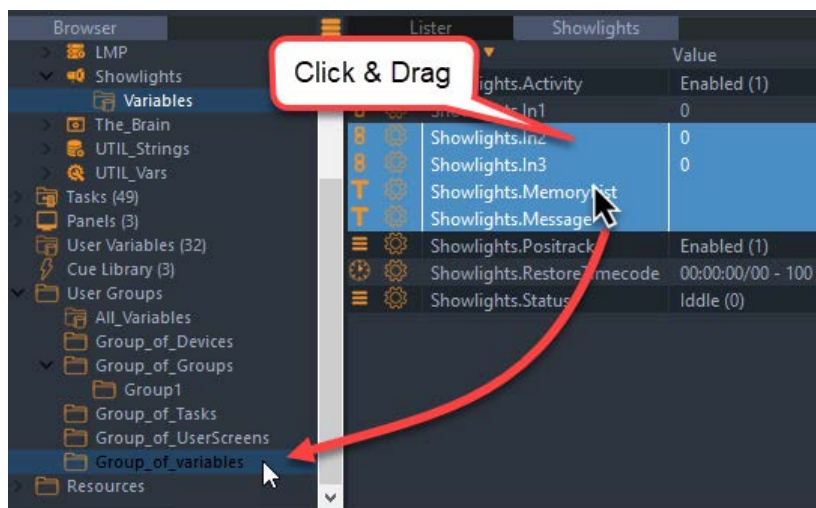
A User Group can contain only one type of object. The first object added to a User Group defines the type of the User Group.

Adding Objects From the Lister

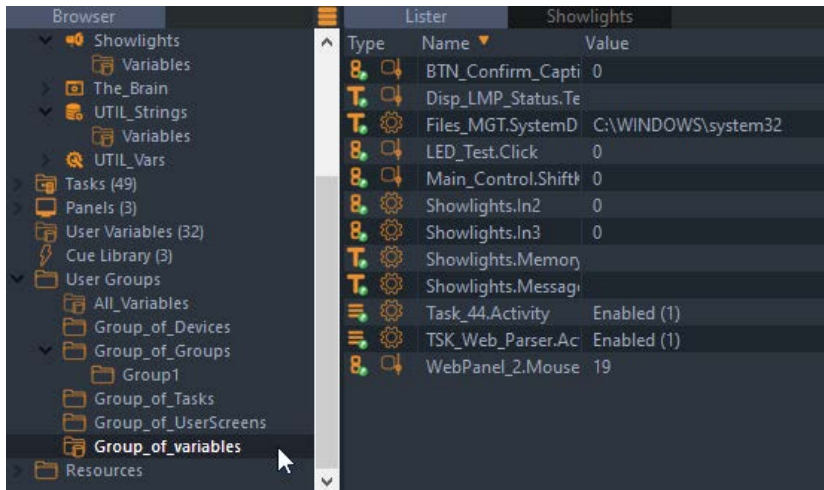
The easy way to add objects to a User Group is to select the objects in the Lister then drag and drop them into the User Group.

For example, to create a User Group for variables:

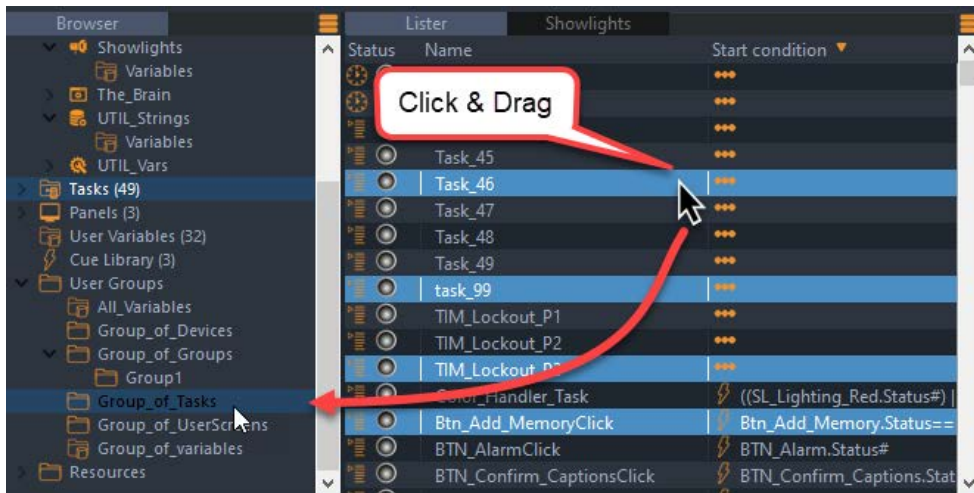
Select some variables and drag and drop them into the User Group.



Clicking the User Group in the Browser tab displays a filtered Object Lister.

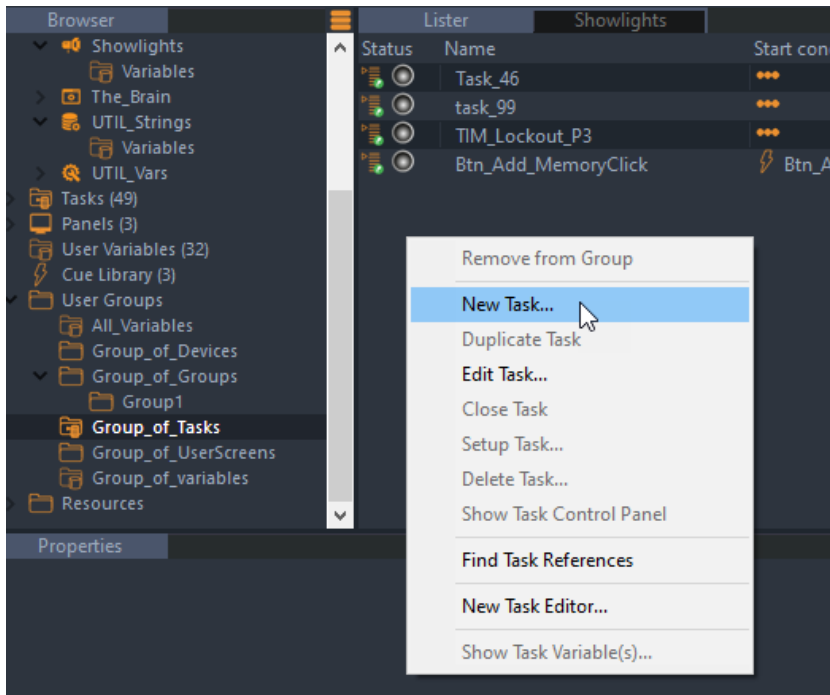


Another example, with tasks:



Adding New Objects Within a User Group

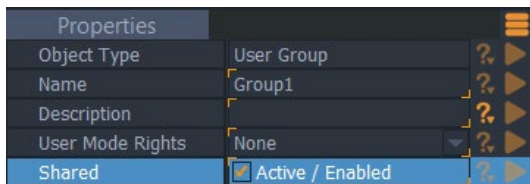
You can create new objects in your project from within a User Group. The new objects are created in their proper object node and added to the User Group from which you created them.



Object creation depends of the type of the User Group (tasks, variables, UserScreen, devices).

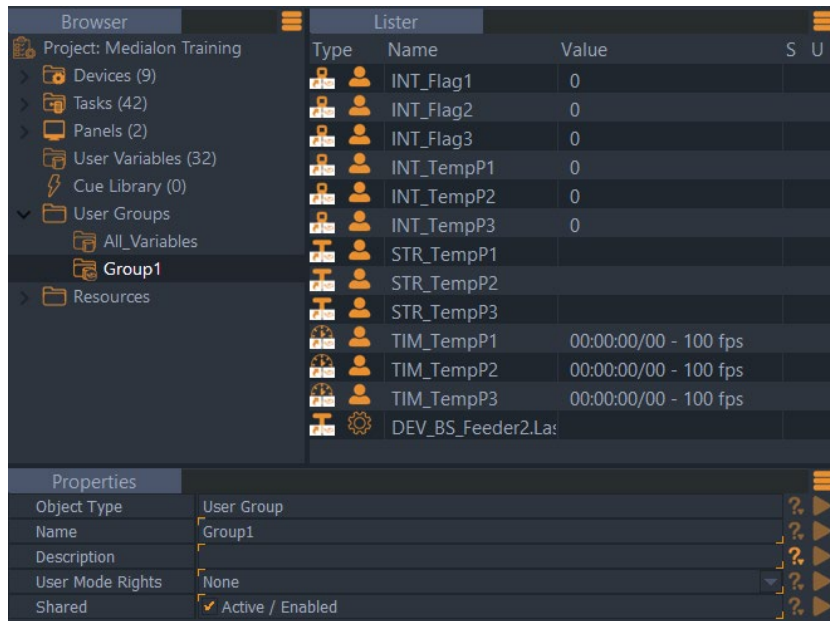
7.2.3 Sharing Objects

If you wish to exchange information and commands with other Medialon Manager V7 Projects or other software using the OpenCap API, activate the Shared option in the Properties of the User Group.



Shared groups are used by Medialon Manager to publish variables, tasks, and UserScreens on the network.

Only the variables, tasks, and UserScreens contained in a Shared User Group will be available for commands from Medialon On Network devices.



7.2.4 Smart Groups

Devices, tasks and variables have two new properties, Tags and Color Tags, that enable automatic sorting into Smart Groups. Multiple Tags can be used in each device, task or variable allowing the ability to be sorted into multiple Smart Groups easily. To add a Tag, click on the wand icon to the right of the Tags property. Then type a new tag in the edit window and click the + to add it to the All Tags list. Double click on a selection in the All Tags list to assign it to the Object Tags list.

Color Tags provide a color 'chip' in the Lister instance of that object making it easier to spot when scrolling through a list.

Right-click on the User Groups node in the Browser tab to create a new Smart Group. Once created, each Smart Group can be configured for a single Color Tag and/or a single or multiple Tags

- Strict Match – Allows the Smart Group to accept any or all of the specified Tags or Color Tag

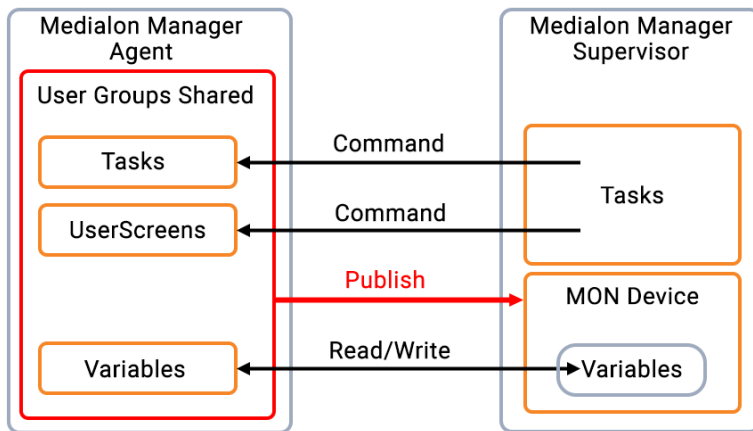
7.3 Networking

When Medialon Manager is connected to a network, it acts as a server and can share its objects with external clients.

7.3.1 Medialon On Network (MON)

Medialon On Network (aka MON) devices in Medialon Manager projects have this capability, but with OpenCap it has been expanded as an Application Programming Interface (API).

Medialon Manager shares tasks, UserScreens and variables like a server. Another Manager project can connect to those shared objects by creating a Medialon On Network Device which acts as a client to the shared User Groups in the other project.

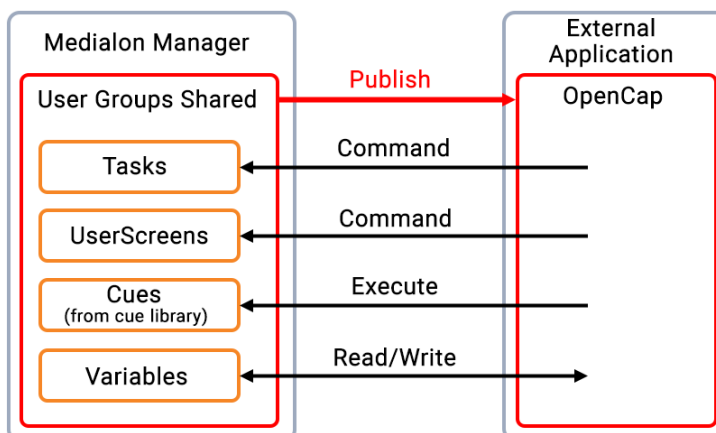


Shared variables are read/write on both Medialon Manager stations. The Client can send commands across the network to shared tasks or UserScreens on the Server.

Medialon Showmaster can share tasks, variables, and UserScreens. Another Medialon Manager can detect a Showmaster's shared objects.

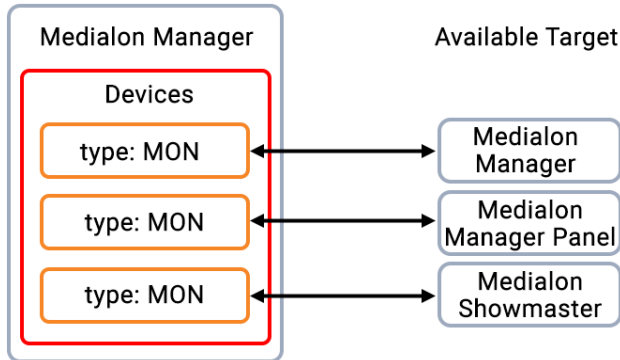
7.3.2 OpenCap

External applications can access Medialon Manager objects using Medialon's OpenCap API, which is available as an API. The API is available as an XML protocol, communicated via a TCP session, or it is available as a DLL (Dynamic Link Library) that connect to Medialon Manager using the binary OpenCap protocol, allowing developers to read data and trigger tasks or change the value of variables in their Medialon Manager project. The API offers several programming examples in different languages (C++, Visual Basic, C#). Contact support@7thsense.one for further information.



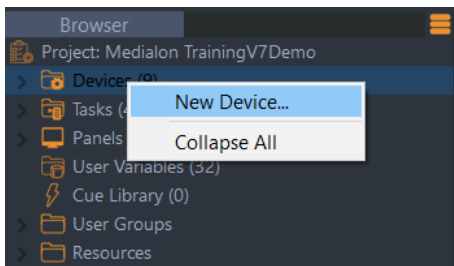
7.3.3 Medialon On Network Device

Medialon Manager V7 can control another Medialon target station. A 'Medialon On Network' (MON) device must be created. The Medialon project that has the MON device acts as Supervisor (Client) of the Agent (Server) station. It has access to shared objects, it sends commands to the Agent, and it monitors the Agent's variables.

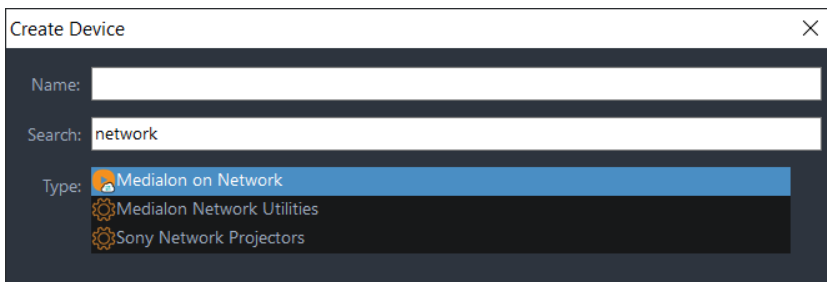


Creating a device

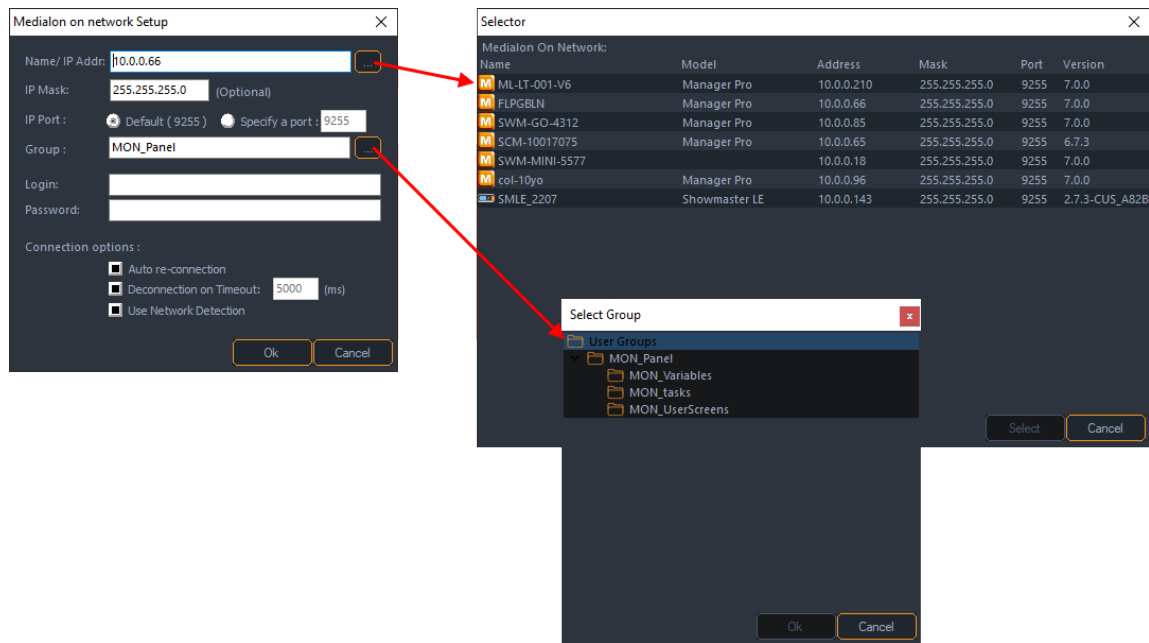
1. Select the Devices item in the Object Browser and use the right mouse click create a new Device:



2. Select 'Medialon on Network' type.
3. Name the Device:

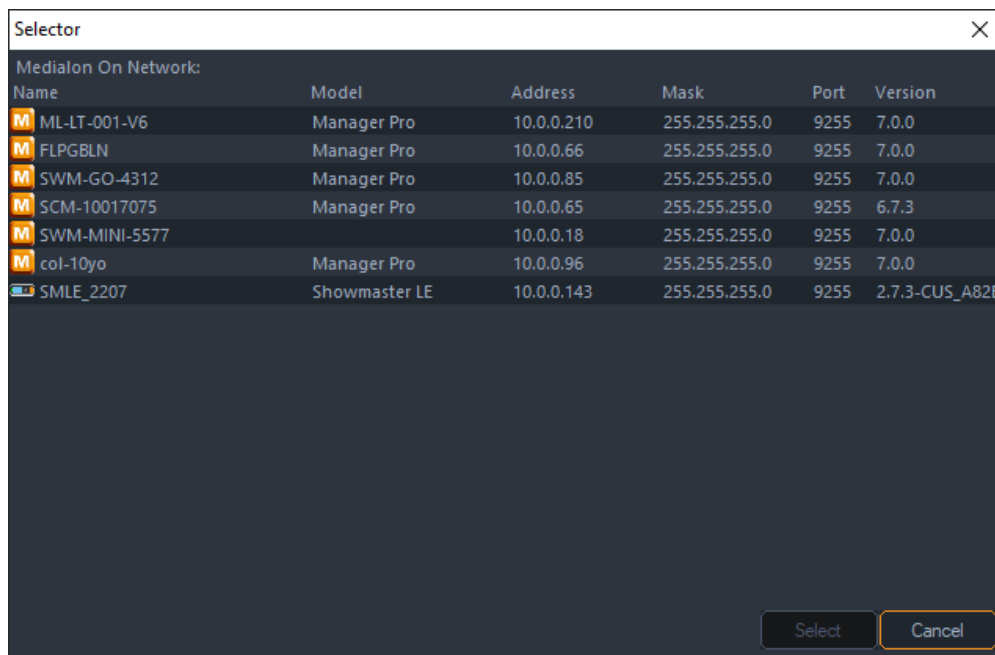


4. The Device Setup dialog box opens:



Name/IP Addr:

Specifies the computer name or the IP address of the Agent target. A wizard can scan the network and find all Medialon targets present (Medialon Manager or Showmaster).



IP Mask (Optional):

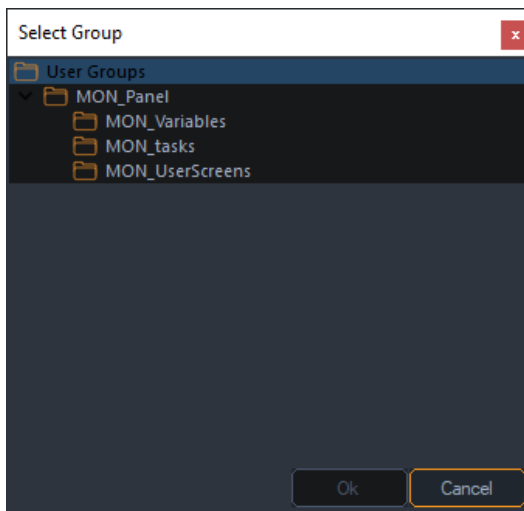
Specifies an IP Mask to reach the Agent if the Agent is on a different sub network.

IP Port:

Defines the IP port number used by the Medialon Agent(s) to listen for connections from the Medialon On Network Device. The default IP port number is 9255.

Group:

Select a Shared Group of objects (tasks, variables, UserScreens) shared by the Agent. Use the wizard button to get the list of available groups.



Login/Password:

Medialon On Network connections can be protected by login credentials. Enter, if needed, the login and password required by the selected Agent.

Auto re-connection:

If checked, the Medialon On Network Device automatically attempts to reconnect to the Agent if the connection is lost.

Disconnection on timeout:

Specifies the amount of time that needs to pass with no response from the Agent for the MON device to determine the connection has become disconnected. When this occurs, the TCP session is closed and the Auto reconnection setting determines whether the MON device attempts to reconnect automatically.

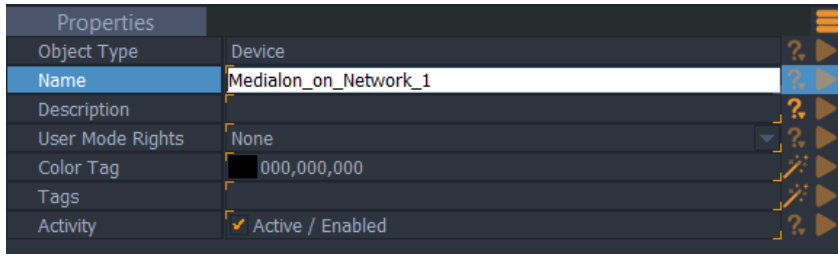
Use Network Detection:

Enables a UDP network detection of the Agent prior to attempting a TCP connection. It prevents the Microsoft Windows 'anti hacker' security policy from being triggered when multiple TCP connections are attempted but the devices do not exist on the network. This setting attempts to determine if the Medialon Agent is active before attempting to establish the TCP connection.

5. Once you have finished configuring the device, click 'OK'. The new device is created and appears in the Browser tab and Device Map tab.

Device Properties

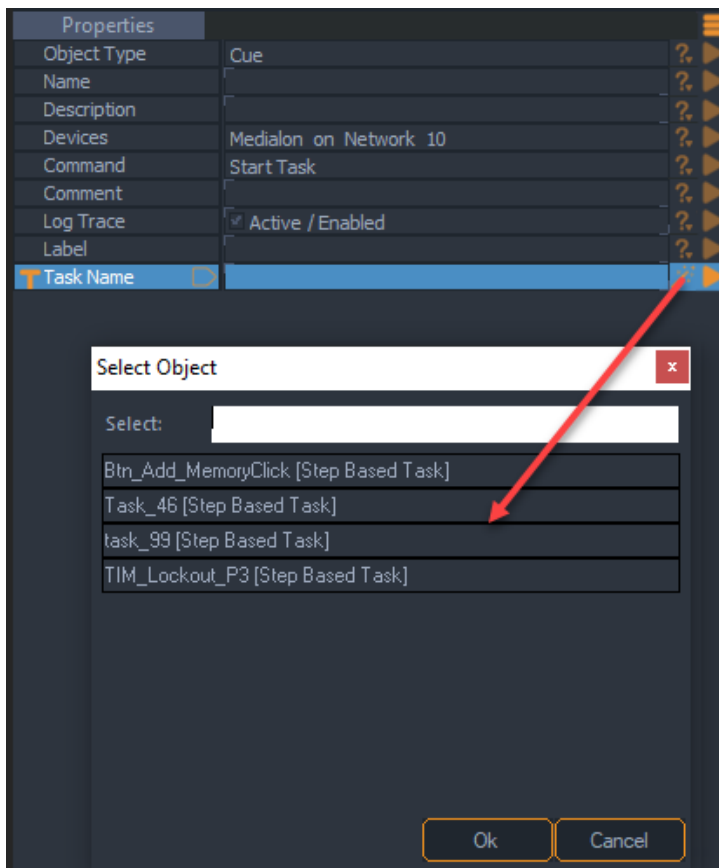
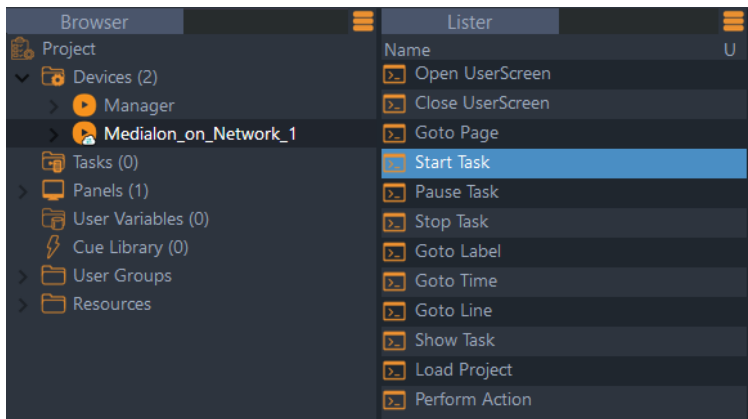
Select Medialon On Network Device in the Browser tab or Device Map tab. Its properties are displayed in the Properties tab.



You can rename the Device. Remember to validate your changes with the *Enter* key.

Device Commands

Medialon Manager can send commands to Medialon On Network Devices.



These commands are performed on the shared tasks and UserScreens. Use the wizard to select available objects from the Agent project in the cue properties.

Load Project

The 'Load Project' command loads another Medialon Manager V7 project on the Agent system (not available for Showmaster). You can choose what mode to run the Agent's project.

Parameters:

- *Project Name*: [Type: String]: path and name of the project to be opened
- *Run mode*: [Type: Enum]: Select the mode in which the project starts. Available values:
 - *Project Autorun*: Uses the project's autorun preferences settings
 - Stop
 - Debug
 - User
 - Run



Note that the newly loaded project must use the same Shared Group as the original Agent project. If not, Medialon Manager V7 will not be able to reconnect to the Shared Group that was configured in the MON device's setup, and it will lose access to the variables, tasks, and UserScreens of the new project file.

Open UserScreen

This command allows the Supervisor to open a UserScreen on the Agent project.

Parameters:

- *UserScreen Name*: [Type: String]: Name of UserScreen to be opened; use the wizard to select an available UserScreen from the Agent project.
- *Position*: [Type: Enum]: Position mode, available values:
 - *Default*: Uses the default UserScreen position; does not use the Left and Top parameters
 - *Absolute*: UserScreen position is given by Left and Top parameters
- *Left*: [Type: Integer]: Left position of UserScreen
- *Top*: [Type: Integer]: Top position of UserScreen

Close UserScreen

The opposite of the previous action; closes a UserScreen in the Agent project.

Parameters:

- *UserScreen Name*: [Type: String]: Name of UserScreen to be closed; use the wizard to select an available UserScreen

Goto Page

Switch to the desired page of the UserScreen.

Parameters:

- *UserScreen Name*: [Type: String]: Name of UserScreen to switch pages on; use the wizard to select an available UserScreen
- *Page*: [Type: Integer]: Page number to go to

Start Task

Launch a task in the Agent project.

Parameters:

- *Task Name*: [Type: String]: Name of task to start; use the wizard to select an available task

Pause Task

A task in the Agent project may be suspended momentarily and then continued again from where it was paused. To restart a task, use the 'Start Task' command.

Parameters:

- *Task Name*: [Type: String]: Name of the task to pause; use the wizard to select an available task.

Stop Task

This command serves not only to stop a task, but to take it back to the start, so that the next 'Start Task' command makes it restart from the beginning.

Parameters:

- *Mode*: [Type: Enum]: Available values:
 - *Specified Task*: Stops the task in the Agent project specified in the Task Name property.
 - *All Tasks (Inclusive)*: Stops all tasks in the Agent project.
 - *All Tasks (Exclusive)*: Stops all tasks in the Agent project (there is no difference between the two Stop All Tasks commands).
- *Task Name*: [Type: String]: Name of task to stop. Use the wizard to select an available task.

Goto Label

A 'cue' is identified by a number (in the case of step based tasks) or a time (for time based tasks). A 'Label' may be added to any cue. The advantage of a 'Label' is that it is associated with the cue and thus cannot be changed as a cue index can in the event of an insertion, or as the time can shift in the event the cue is moved on the timeline.

The purpose of the 'Goto Label' command is to reach the specific cue associated with the Label and to launch the execution of the task from this Label. The same Label may be used in several different tasks.

Parameters:

- *Task Name*: [Type: String]: Name of the task to go to label; use the wizard to select an available task.
- *Label*: [Type: String]: Label to go to in the task.

Goto Time

This command is exclusive to time based tasks and enables you to reach a given time in a time based task and then to launch its execution starting at this time.

Parameters:

- *Task Name*: [Type: String]: Name of the task in the Agent project to go to time; use the wizard to select an available time based task.
- *Time*: [Type: Time]: Time to go to in the task.

Goto Line

This enables you to access a 'cue' index in a step based task in the Agent project and to launch the execution from this 'cue'.

Parameters:

- *Task Name*: [Type: String]: Name of the task in the Agent project to go to line number; use the wizard to select an available step based task.
- *Line*: [Type: Integer]: Line to goto in the task.

Show Task

Opens or closes a task in a Task Editor window in the Agent project.

Parameters:

- *Task Name*: [Type: String]: Name of the task to show or hide; use the wizard to select an available task.
- *Status*: [Type: Enum]: Show or hide, Available values:

- *Show*: The Task Editor is opened
- *Hide*: The Task Editor is closed
- *Left*: [Type: Integer]: Left position of the Task Editor window
- *Top*: [Type: Integer]: Top position of the Task Editor window
- *Width*: [Type: Integer]: Width of the Task Editor window
- *Height*: [Type: Integer]: Height of the Task Editor window

Device Variables

A shared variable is available to the other Medialon Manager systems that are connected with Medialon On Network devices. Variables are detected when both applications are in Debug, User or Run Mode. Shared variables can be used by any remote station just like a local variables.

Type	Name	Value	S
☰ ⚙	Medialon_on_Network_10.Activity	Enabled (1)	
8 ⚙	Medialon_on_Network_10.BTN_Confirm_Captions.Statu	0	
T ⚙	Medialon_on_Network_10.Disp_LMP_Status.Text		
T ⚙	Medialon_on_Network_10.Files_MGT.SystemDirectory	C:\WINDOWS\system3.	
8 ⚙	Medialon_on_Network_10.LED_Test.Click	0	
8 ⚙	Medialon_on_Network_10.Main_Control.ShiftKeyStatus	0	
8 ⚙	Medialon_on_Network_10.Showlights.In2	0	
8 ⚙	Medialon_on_Network_10.Showlights.In3	0	
T ⚙	Medialon_on_Network_10.Showlights.MemoryList		
T ⚙	Medialon_on_Network_10.Showlights.Message		
☰ ⚙	Medialon_on_Network_10.Status	Not Connected (0)	
☰ ⚙	Medialon_on_Network_10.Task_44_Activity	Enabled (1)	

The list of variables depends on the variables group shared by the Agent project.

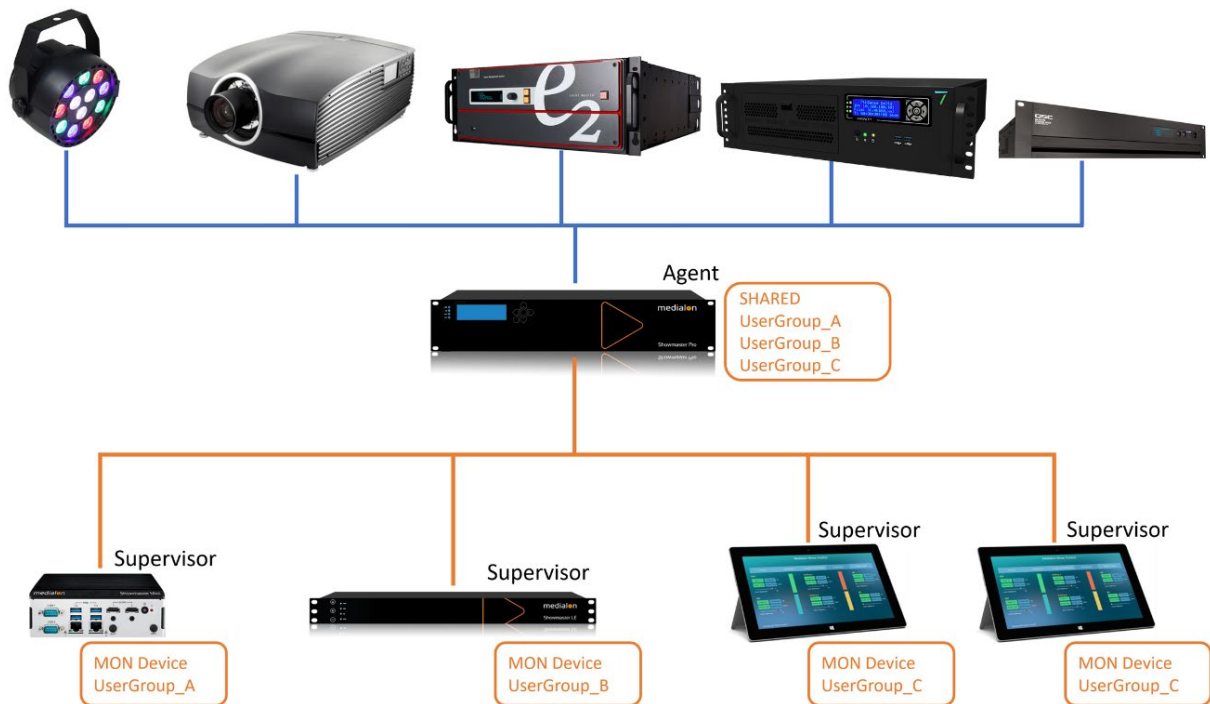
- *Medialon_On_Network.Status*: variable reflects Device connection status:
 - *Not Connected*: No Medialon On Network station was found.
 - *Connected*: Medialon On Network connection is established.
 - *Network Error*: Medialon Manager could not find the Agent.
 - *Identifying*: Connection is opened, Medialon Manager is sending authentication.
 - *Connection Denied*: Connection is rejected, bad User name or Password.
 - *Sensing*: Medialon Manager trying to detect the Agent.
 - *Connecting*: Medialon Manager trying to open a connection.
 - *Limited Connection*: The Group name in the Medialon on Network device does not match an available Shared User Group in the Agent project.

7.3.4 Network Architectures

These configurations serve as examples; other configurations may be set up depending on your needs.

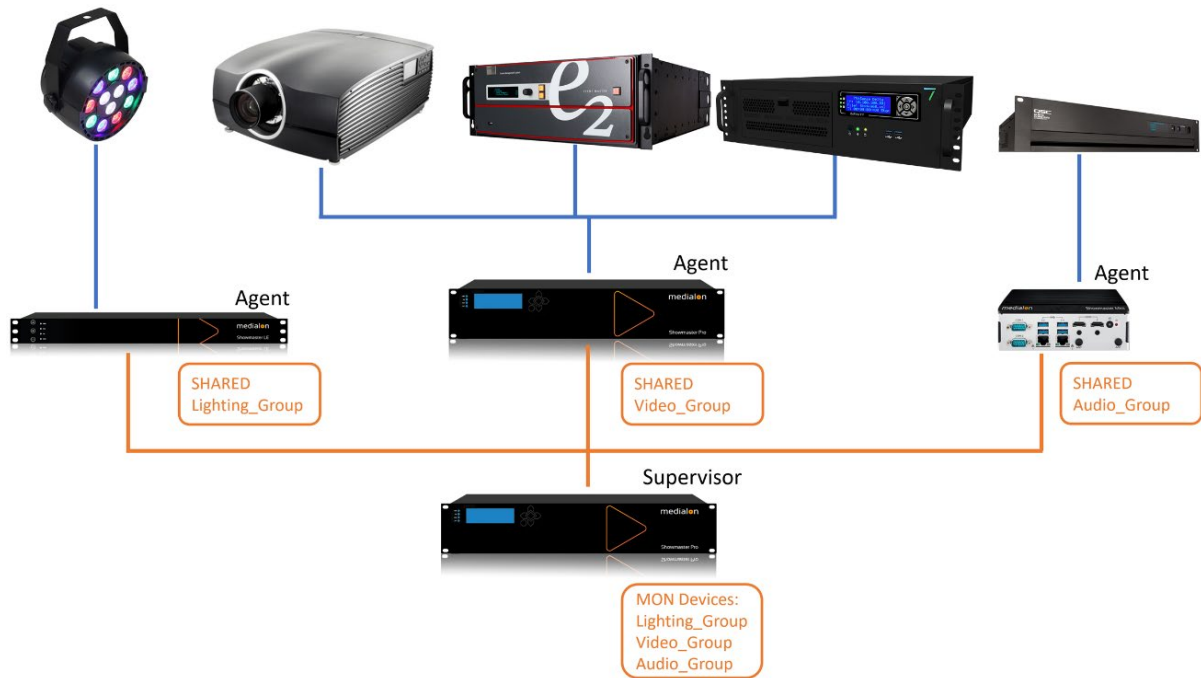
Single Agent with One or More Supervisors

In a single Agent network architecture, one Medialon system shares its variables and accepts network Start Task Commands from one or more Supervisor Medialon systems on the Network. This topology is used if a show needs a 'device server' and separate 'logic servers' or multiple touch panels for interaction with the devices.



Multiple Agents

In a Multiple Agent network configuration, the Supervisor system monitors the Agents' shared variables (read status of shows and devices) and can start local tasks on the Agents. This topology is common in supervisory systems where one Supervisor can monitor the status of many systems (attractions, exhibits, etc.)



7.3.5 Medialon Manager Panel

Medialon Manager Panel is a remote control software for Medialon Manager.

Manager Panel is based on Medialon Manager V7 software and is therefore completely compatible with Medialon Manager. There are, however, the following limitations:

- MxMs: Manager Panel does not support MxMs and therefore does not allow you to communicate with any external devices other than Medialon On Network devices.
- Tasks: Manager Panel can only manage step based tasks; no timelines are available.

Panel Connections

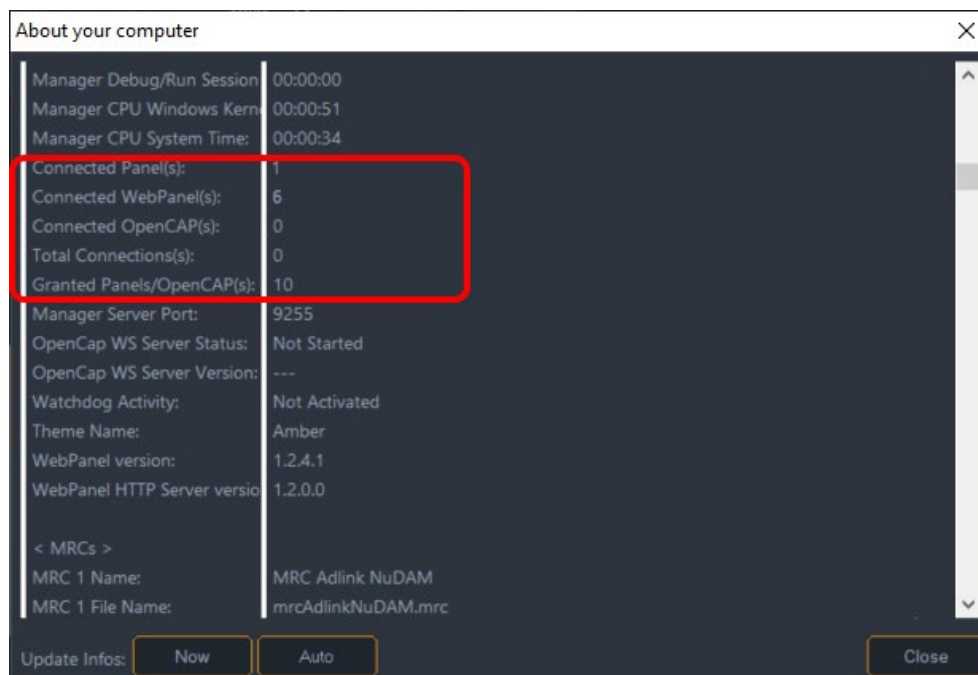
- Showmaster LE can be connected to three Manager Panels
- Showmaster Pro can be connected to ten Manager Panels
- Manager V7 Lite can be connected to two Manager Panels
- Manager V7 Pro can be connected to ten Manager Panels

7.3.6 Panel/WebPanel/OpenCap Connection Chart

No. of connections	Showmaster V2 LE	Showmaster V2 Pro	Manager V7 for Showmaster Mini	Manager V7 Lite	Manager V7 Pro	Manager V7 Panel
Showmaster V2 LE	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited	3 **
Showmaster V2 Pro	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited	10 **
Showmaster Mini	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited	1
Manager V7 Lite	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited	2 **
Manager V7 Pro	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited	10 **
Web Panel	3 **	10 **	1	2 **	10 **	
Manager V7 Panel						
OpenCAP API						

** Number of connections is shared between WebPanel, Manager Panel and OpenCAP API.

Panel connections can be seen in Help > About Your Computer:



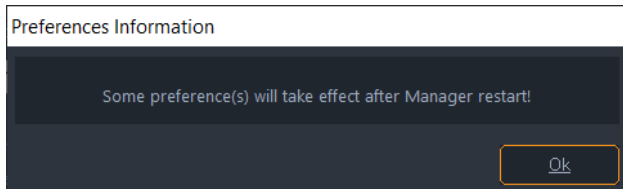
Medialon Manager V7 granted connection can be increased by batch of 10 connections, contact your Medialon dealer.

Medialon Manager V7 Panel can have two MON devices, regardless of what targets they are connected to. Manager Panel cannot connect to another Manager Panel instance.

7.4 Preferences

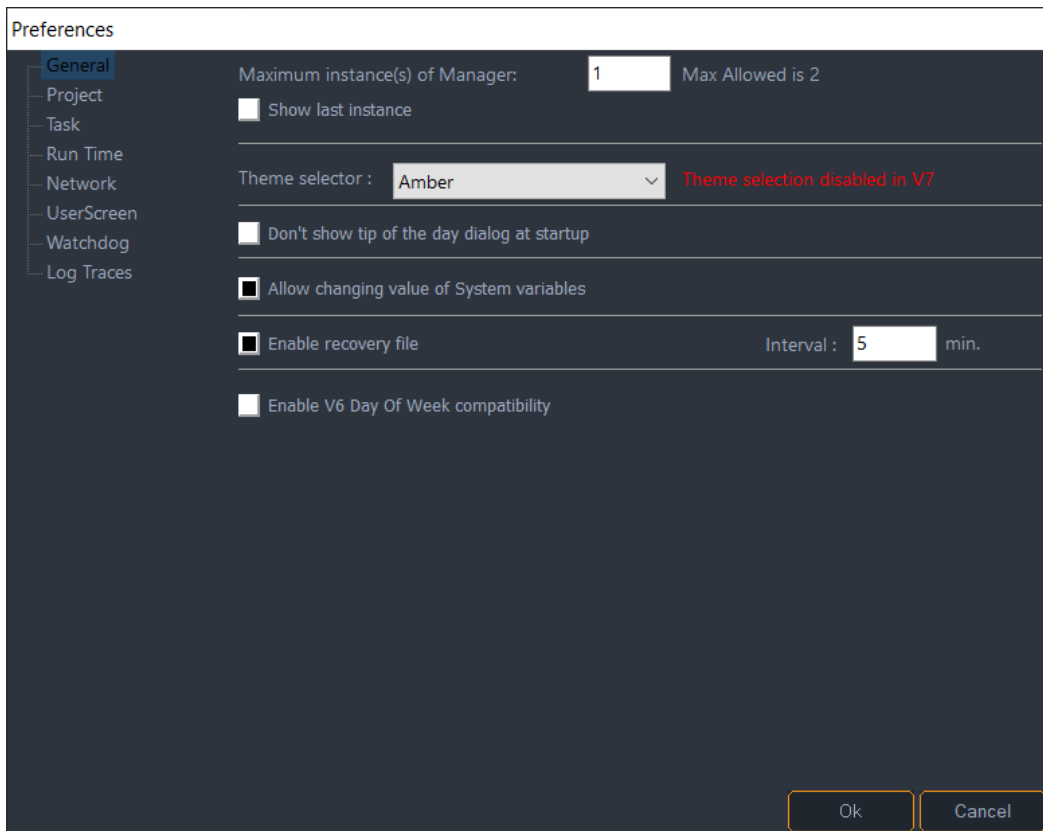
Medialon Manager V7's behavior and the current project behavior can be customized to fit your needs via the Preferences window. Open this from the File > Preferences menu. or right-click on the Project name in the Browser tab.

Some changes will only take effect after restarting Manager:



7.4.1 General

These settings are about the general Medialon Manager behavior and are not specific to the current project.



Maximum Instance(s) of Manager

Enables you to determine the number of times the Medialon Manager V7 application can be opened (started) simultaneously on the same machine.

Setting this option for 1 'Maximum instance of Manager' allows only one Medialon Manager application to be open at any one time; this will be displayed on the foreground each time that an attempt to open another instance is made.

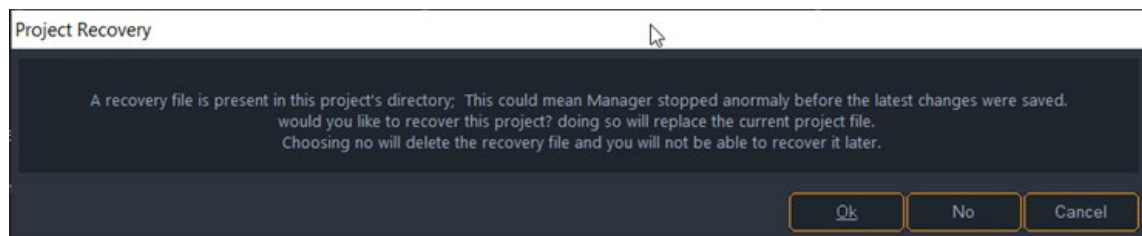
Allow changing value of System variables

For debugging purpose, you may need to manually change variable value of devices, especially if you work 'offline'. This setting can be edited in the general tab in preferences.

Enable recovery file

NEW: This enables an auto-save function with a selectable duration. Select the required interval in minutes.

The next time you load a project with autosave data associated with it, Manager V7 asks if you want to load the autosaved data:



Enable V6 Day of Week compatibility

Changes the Manager.CurrentDate variable's Day of the Week enumeration to match Manager V6 style.

- V6 style: Manager.CurrentDate shows Monday (2)
- V7 style: Manager.CurrentDate shows Monday (1)

7.4.2 Project

Defines Medialon Manager V7's current project information.

Preferences

- General
- Project**
- Task
- Run Time
- Network
- UserScreen
- Watchdog
- Log Traces

Title : Main auditorium

Author : Starr Tupp

Subject : Medialon Control System

User defined loading splash bitmap : splash.bmp

Notes:
Enter additional information for support and programmers

Ok Cancel

Title

Provide a title for the project. The Title is displayed in the default Medialon Manager loading splash window and in the Import templates menu.

Author

This is an information field for the project.

Subject

This is an information field for the project.

User defined loading splash bitmap

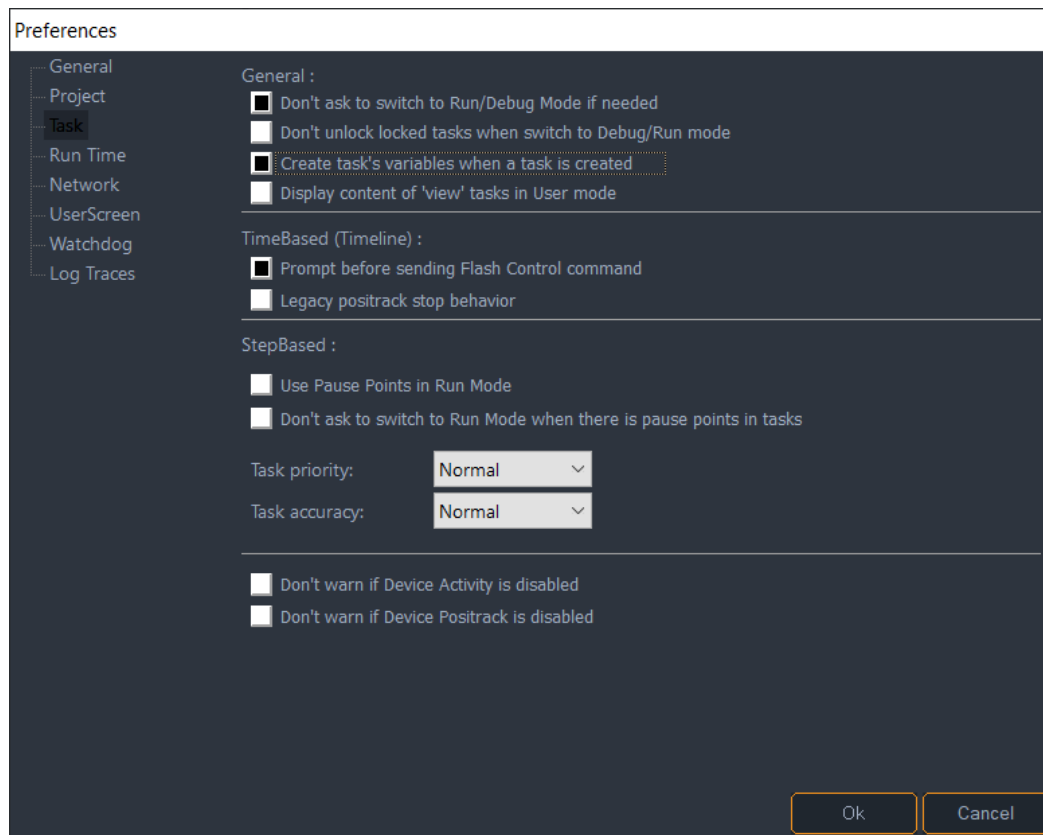
Select a custom bitmap (.bmp) to be displayed on the loading splash window instead of the default one. A bitmap is selected either by double-clicking in the image preview field or in the bitmap file name field. This function is only available once the New Project has been created and saved once.

Notes

This text area can be used to add extra information about your project or history of main change.

7.4.3 Task

Task preferences defines task behavior.



General

Don't ask to switch to Run/Debug Mode if needed

Enables or disables the user prompt to confirm the switch to Run or Debug mode from Stop mode when needed.

Don't unlock locked tasks when switch to Debug/Run mode

Locked tasks are automatically unlocked when switching to Debug or Run mode unless this is checked.

Create task's variables when task created

Enables or disables the automatic creation of task variables when the task is created. This sets the default state of variable creation check box in the Create Task dialog box.

Open content of 'View' Tasks in User Mode

Enables or disables the ability to open tasks in a read-only Editor when a task's User Mode Rights are set to View, when in User Mode.

Time Based (Timeline)

Prompt before sending Flash Control command

Enables or disables the confirmation prompt when a Flash command is about to be sent (when a user clicks on a Flash command button). See also [Flash Controls](#).

Legacy positrack stop behavior

Switches Positrack Stop to legacy behavior. See also [Positrack](#).

StepBased

Use Pause Points in Run Mode

Enables or disables the ability to use Pause Points in Run Mode. This option is unchecked by default and it is strongly recommended not to use this option except for specific purposes as tasks could get 'stuck' and unable to advance.

Don't ask to switch to Run Mode when there are pause points in tasks

Enables or disables the confirmation prompt when switching to Run mode when Pause Points are present in the project. This option makes sense only if the 'Use Pause Points in Run Mode' option is checked and it's strongly recommended to enable this option in this case.

Task priority

Determines the priority of execution of all the step based tasks in relation to the other Medialon Manager V7 modules and any other applications. Execution priority stands for the extent to which the choice of precision for step based tasks is to be respected. The higher the priority, the more the selected precision will be respected. For most purposes, leave this set to Normal.

Task accuracy

Determines the execution precision of the entire Medialon Manager V7 step based tasks engine.

The precision influences the rapidity of execution of the commands present in step based tasks. The higher the precision, the more precisely and quickly the tasks are executed. But increasing the precision also increases the load on the machine's processor, so it must be adjusted according to the overall use of the processor by other applications. For most purposes, leave this set to Normal.



Time based tasks are not affected by these parameters since they are based on a very precise time reference that is independent of the 'step based' tasks.

Don't warn if Device Activity is disabled

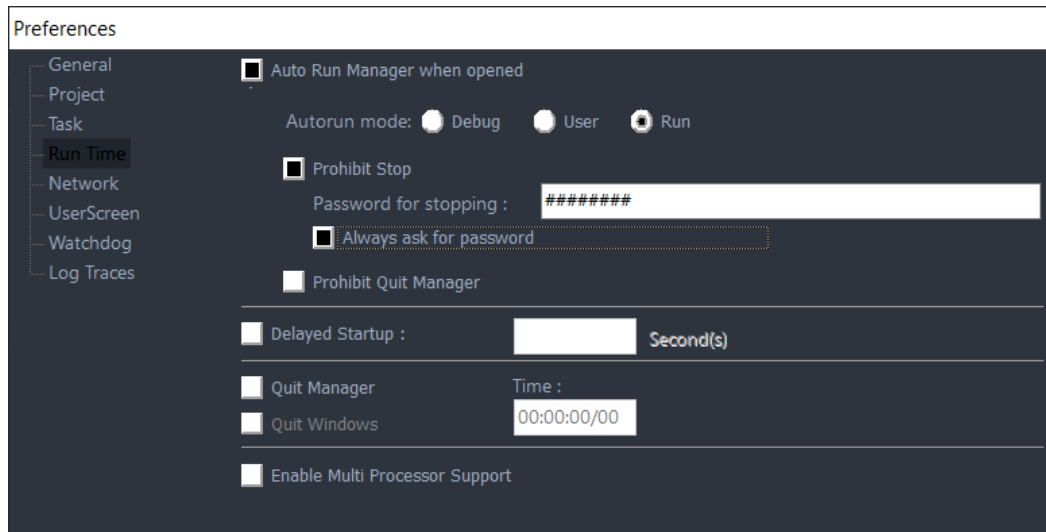
Enables or Disables the user prompt to warn when a device activity is disabled.

Don't warn if Device Positrack is disabled

Enables or Disables the user prompt to warn when a device Positrack is disabled.

7.4.4 Run Time

Organizes the automation of the opening of Medialon Manager, of its automatic start-up, of the end of its operation and automatic shutdown. The halting of the execution of a project may also be inhibited or protected by a password.



Auto Run Manager when opened

Determines if Medialon Manager V7 should launch in Run/User/Debug mode when a project is opened. This option only applies when Medialon Manager is opened by via a project, by double-clicking on a Project File icon or by placing a shortcut to the project in the *Start > All Programs > Startup* folder, for example.



We strongly recommend registering your license if you plan to start your project with the 'autorun' option. The registration dialog box will continue to be displayed each time you launch your project. See also [Registration](#).

Autorun mode

Determines which mode to switch to if the Autorun option is checked. The *.mngProj shortcut can be copied into the Windows startup folder to automatically launch it when Windows starts up.

Prohibit Stop

Requires a password before switching from Debug/User/Run to Stop mode.

Always ask for password

Enables or disables asking the user for a password even if Medialon Manager was manually started.

Prohibits Quit Manager

Enabling this option makes it impossible to quit Medialon Manager using standard commands; *Alt+F4*, for example.

Delayed Startup

Defines, in seconds, the time delay at startup between the moment Medialon Manager is launched and the moment it becomes active. This can be useful in certain configurations, for example when Medialon Manager could potentially start before the License Key is detected by the system.

Quit

Quit Manager

Defines a time at which Medialon Manager will automatically quit. This option takes priority over 'Prohibit Quit Manager'.

Quit Windows

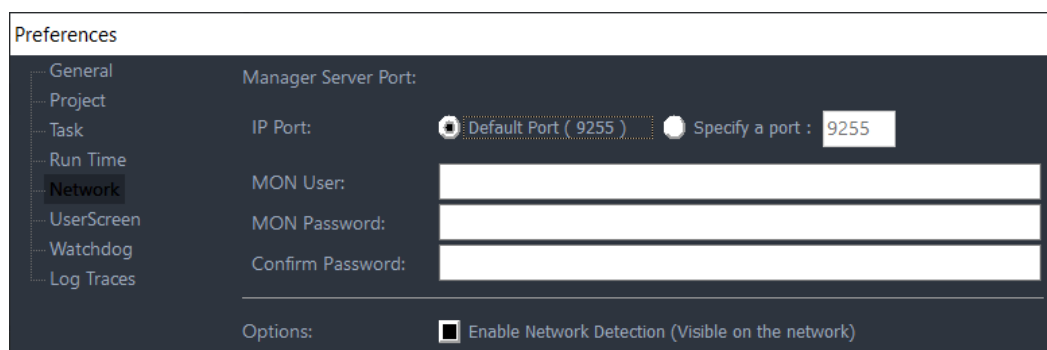
Defines a time at which Medialon Manager and Windows will automatically quit and shut down. This option takes priority over 'Prohibit Quit Manager'.

Enable Multi Processor Support

When this preference is enabled, Medialon Manager distributes its workload between all processors present on your system, instead of allowing Windows to manage processor loads itself.

7.4.5 Network

Defines network settings and Medialon On Network login credentials.



Manager/Showmaster V2 Server Port

Defines the IP port used by Medialon Manager to 'listen' for the connections from Medialon Manager client stations.

IP Port

- Default Port: Selects the default port (9255); in this case the clients must also have the 'Default port' option selected in the 'Medialon On Network' ('Device' setup).
- Specify a port: Allows you to choose a different IP port number.

Medialon On Network Login

These settings set the Agent's login credentials. A MON device connecting to this Agent needs to use the same credentials in the MON device setup.

- MON User: Defines the User login when the project needs to have restricted access.
- MON Password: Defines the User password. It needs to be confirmed.

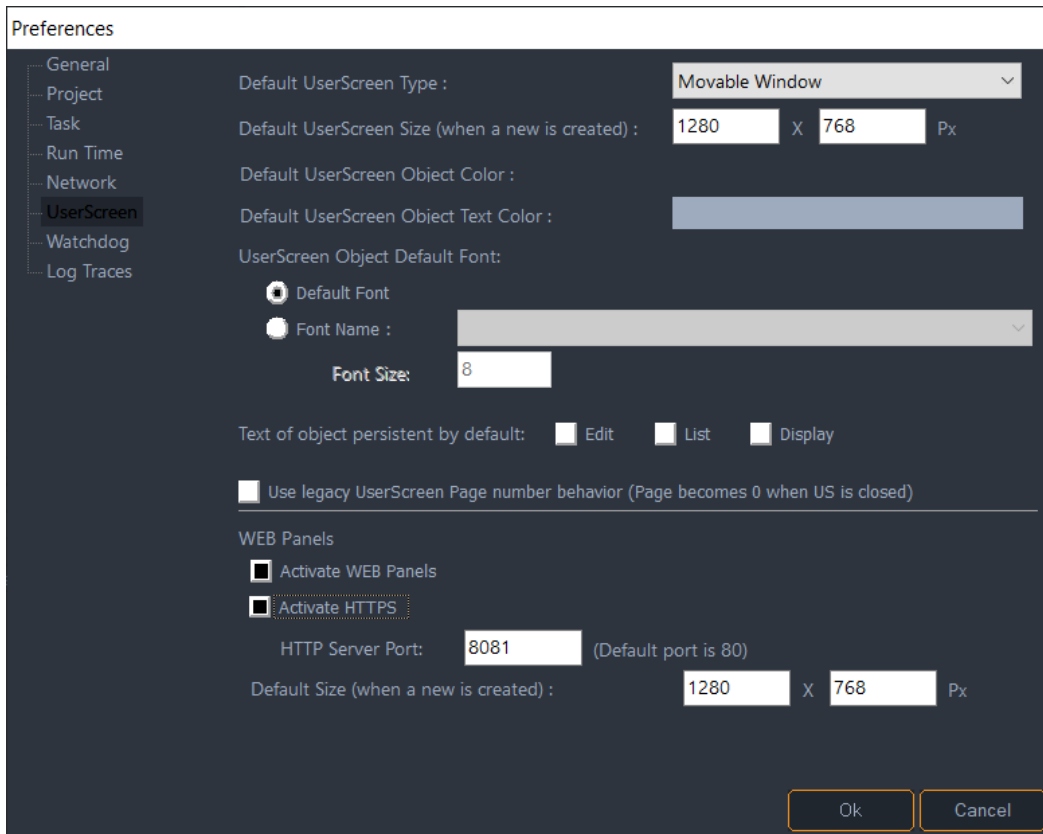
Options

Network Detection

For added security, Medialon Manager implements network detection of Medialon Manager hosts before initiating a connection with the Agent.

7.4.6 UserScreen

Defines default settings for UserScreens, objects and Web Panels used by your project.



Default UserScreen Type

Defines the default type of a UserScreen when it is created.

- Single Window
- Border Window
- Movable Window (Default)
- Full Screen

Default UserScreen Size

Defines the default size of new UserScreens when they are created.

Default UserScreen Object Color

Defines the default color for all dynamic graphical objects and UserScreens.

Default UserScreen Object Text Color

Defines the default color for all text in dynamic graphical objects and UserScreens.

UserScreen Object Default Font

Allows you to select a default font, or specify a particular font to be used when creating an object in a UserScreen that requires text.

Text of object persistent by default

Makes persistent by default the content of text variables related to 'Edit', 'List' or 'Display' objects.

Use legacy UserScreen Page number behavior (Page becomes 0 when US is closed)

When the UserScreen is closed, the first page number becomes 0 rather than 1.

Web Panels

Activate WEB Panels

Enables or disables the WEB Panel HTTP server.

Activate HTTPS

Check to use secure sockets.

HTTP Server Port

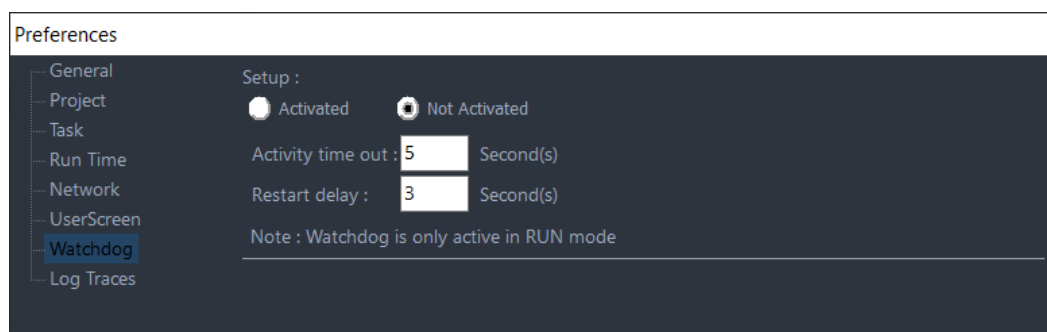
Defines the HTTP port used by Medialon Manager V7 to listen to the WebPanels request, by default HTTP port is 80. If other devices may also be using the default port, use 8081.

Default WebPanel Size

Defines the default size of new WebPanels when they are created.

7.4.7 Watchdog

Watchdog is an external program that regularly checks and controls the proper running of Medialon Manager V7 and can reboot it if it freezes up. See also [Medialon Watchdog](#).



Setup

Activated/Not Activated

Defines whether Medialon Manager uses the services of Medialon Watchdog or not.

Activity time out

Defines the frequency (in seconds) of Manager’s sending of the proper operation signal. If the operational signal is missed, the Medialon Watchdog application will attempt to restart Manager or the PC, depending on what its settings are configured to.

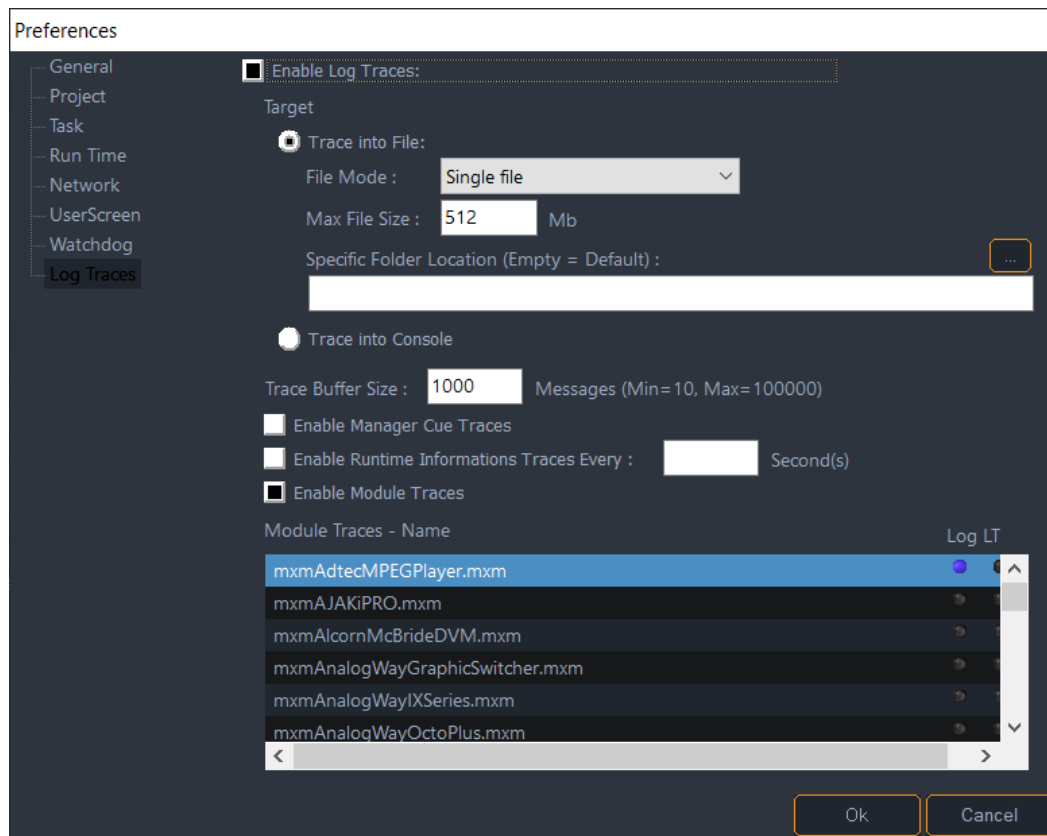
Restart delay

Defines the time (in seconds) that Watchdog must wait before proceeding with a reboot. The actual time before the project is operational will depend on its size.

i Watchdog is active only when Medialon Manager V7 is in RUN mode.

7.4.8 Log Traces

When the Log Traces Option is enabled, Medialon Manager’s activity is traced. This option is only to be used if you are encountering problems or wish to closely follow how your project is running. The log files that are generated can become very large and potentially fill up your hard drive, so this option should not be activated unless there are specific reasons, and should be deactivated when it is no longer necessary.



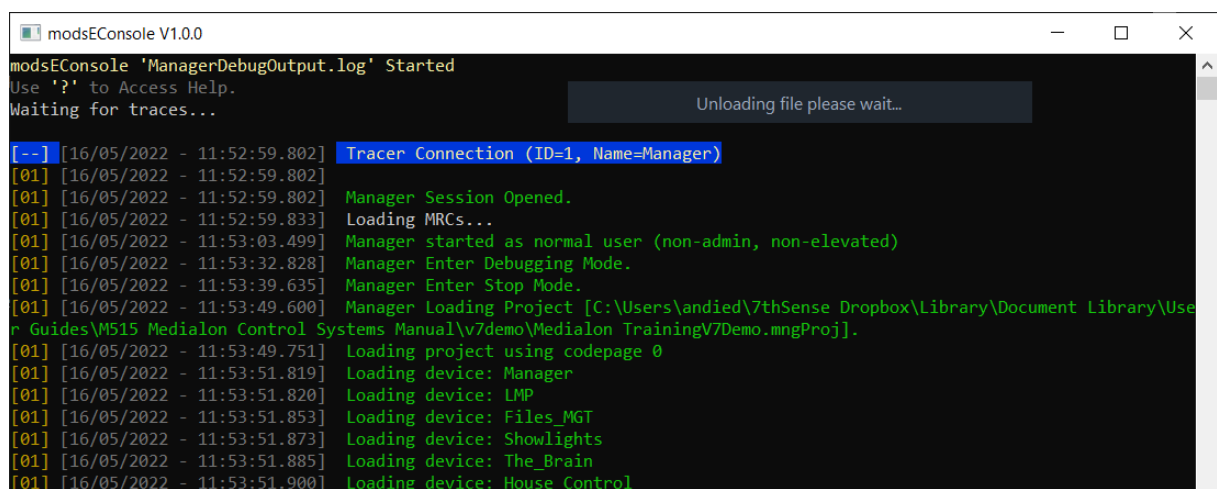
Target

Trace into File

Saves the Trace information into a file, either in one single file, or in a new file every Month, Week, Day, or Session. The maximum size of the generated trace files can be limited by entering the desired value in the 'Max File Size' field.

Trace into Console

Traces information into Medialon Manager's modsEConsole to monitor the activity without creating a log file. This function is used for occasional activity debugging of your project. The modsEConsole opens automatically when Medialon Manager starts if this option is set in the Preferences.



```

modsEConsole V1.0.0
modsEConsole 'ManagerDebugOutput.log' Started
Use '?' to Access Help.
Waiting for traces...
Unloading file please wait...

[--] [16/05/2022 - 11:52:59.802] Tracer Connection (ID=1, Name=Manager)
[01] [16/05/2022 - 11:52:59.802]
[01] [16/05/2022 - 11:52:59.802] Manager Session Opened.
[01] [16/05/2022 - 11:52:59.833] Loading MRCs...
[01] [16/05/2022 - 11:53:03.499] Manager started as normal user (non-admin, non-elevated)
[01] [16/05/2022 - 11:53:32.828] Manager Enter Debugging Mode.
[01] [16/05/2022 - 11:53:39.635] Manager Enter Stop Mode.
[01] [16/05/2022 - 11:53:49.600] Manager Loading Project [C:\Users\andied\7thSense Dropbox\Library\Document Library\User Guides\M515 Medialon Control Systems Manual\v7demo\Medialon TrainingV7Demo.mngProj].
[01] [16/05/2022 - 11:53:49.751] Loading project using codepage 0
[01] [16/05/2022 - 11:53:51.819] Loading device: Manager
[01] [16/05/2022 - 11:53:51.820] Loading device: LMP
[01] [16/05/2022 - 11:53:51.853] Loading device: Files_MGT
[01] [16/05/2022 - 11:53:51.873] Loading device: Showlights
[01] [16/05/2022 - 11:53:51.885] Loading device: The_Brain
[01] [16/05/2022 - 11:53:51.900] Loading device: House_Control

```

Trace options

Enable Manager Cue Traces

This option allows you to enable or disable the tracing of all cues played by Manager. If the program is large and has lots of cues, the log file will quickly become very large. It is therefore preferable to use the Trace into Console option when tracing cue activity.

Enable Runtime Information Traces Every

Should you wish to trace Manager's System information (such as memory footprint) at regular intervals, you can do so by activating this option and determining the interval (in seconds) when the information will be logged.

Enable Module Traces

Enables the tracing of MxMs or MRCs activity within the active session. This function can be useful to trace Network or serial connection problems to external devices.

Configuring Module Traces

MxMs can trace logs into the same file (or console) as Medialon Manager. But MxMs can also trace logs into their own files, even if Log Traces are not enabled in Manager V7's Preferences.

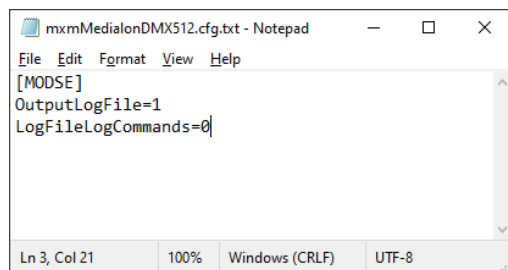
Select modules you need to get the Log traces (Log). If details of commands executed by the module are needed, you can activate Log command Traces (LT).

- Log command Trace option is available with Medialon Manager V6.1.0 or greater.

Settings for log traces of MxMs are only available by editing the configuration file of each MxM and should be carefully used.

Example: (mxmName.cfg, mxmMedialonDMX512.cfg).

This file is a text file, structured like an .ini file, and can be edited with a text file editor like Notepad.



Here are the description of MODSE options you can find in the MxM configuration file Section: [MODSE]

- *Keyname:* OutputLogFile
 - *Description:* Enables log traces using modsEngine
 - *Values:*
 - 0 = Disabled
 - 1 = use Manager settings (default value)
 - 2 = Forced as enabled. If Manager has not activated traces, the MxM will trace into an individual file with a name like mxmName.log (i.e mxmMedialonDMX512.log)
- *Keyname:* LogFileLogCommands
 - *Description:* Defines if the MxM will trace user commands in addition to alarm or error traces.
 - *Values:*
 - 0 = Disabled (default value)
 - 1 = Enabled



Each MxM is provided with a default configuration file.

Example: (mxmName.cfg-dist, mxmMedialonDMX512.cfg-dist)

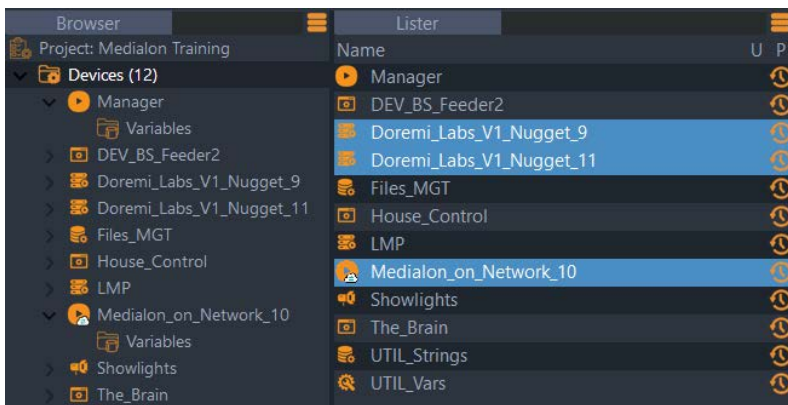
7.5 Advanced Tools

7.5.1 Cue Multiple Devices

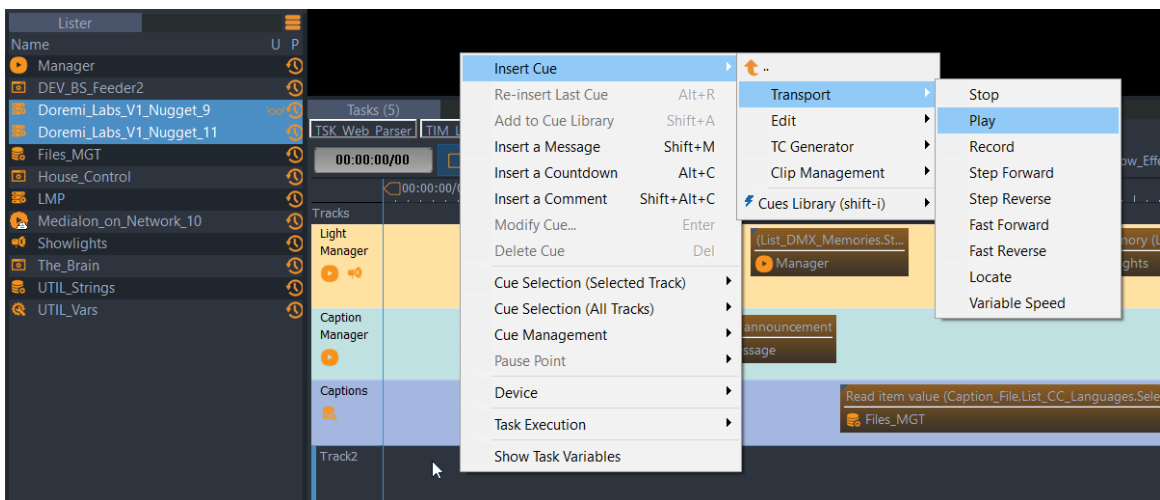
When you need to send the same command to several devices of same type, you can insert one cue for each device, all in the same position, or you can insert one cue linked to all of the controlled devices.

Method to insert one cue for multiple Devices.

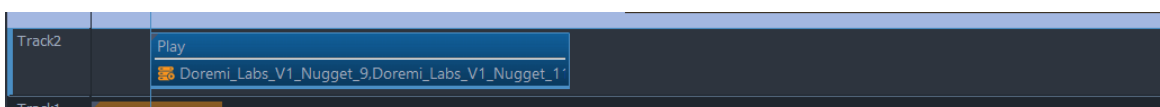
1. In the Browser, click on Devices to display all Devices in the Lister.
2. In the Lister, select Devices of the same type (MxM). Use the *Shift* and/or *Ctrl* keys to add Devices to the selection.



3. Right-click inside the task editor to insert a command. Example on Track 2:



is inserted as this:



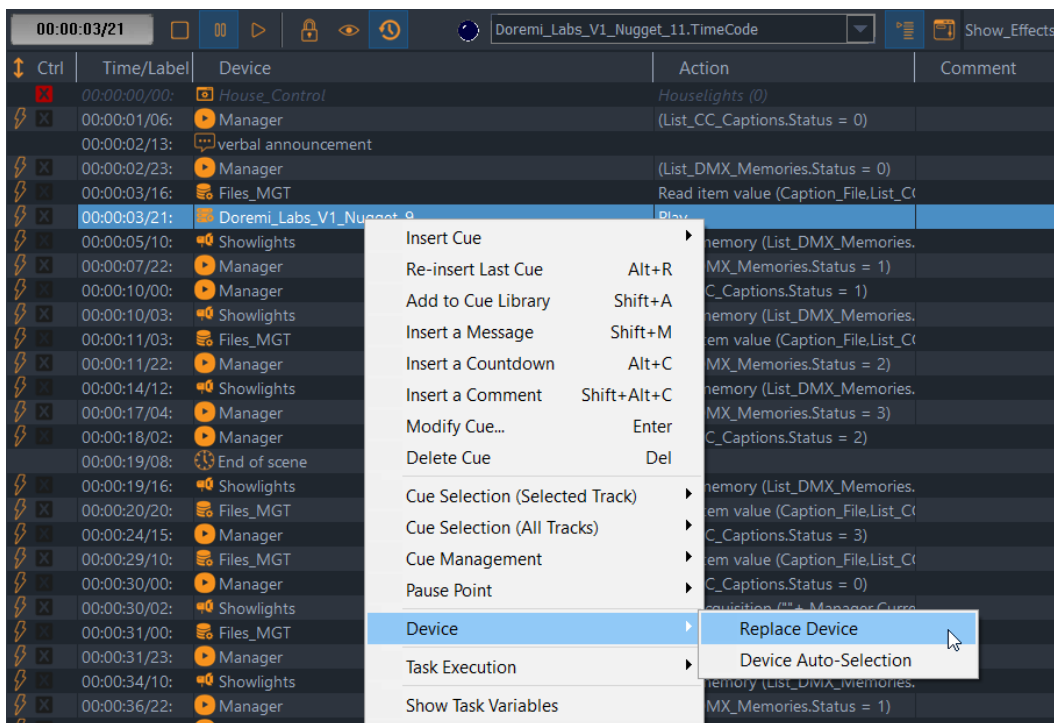
7.5.2 Device Replacement

Medialon Manager V7 integrates a useful function that allows you to reassign a cue to a different device, or replace one device with another.

Device Cue Reassignment

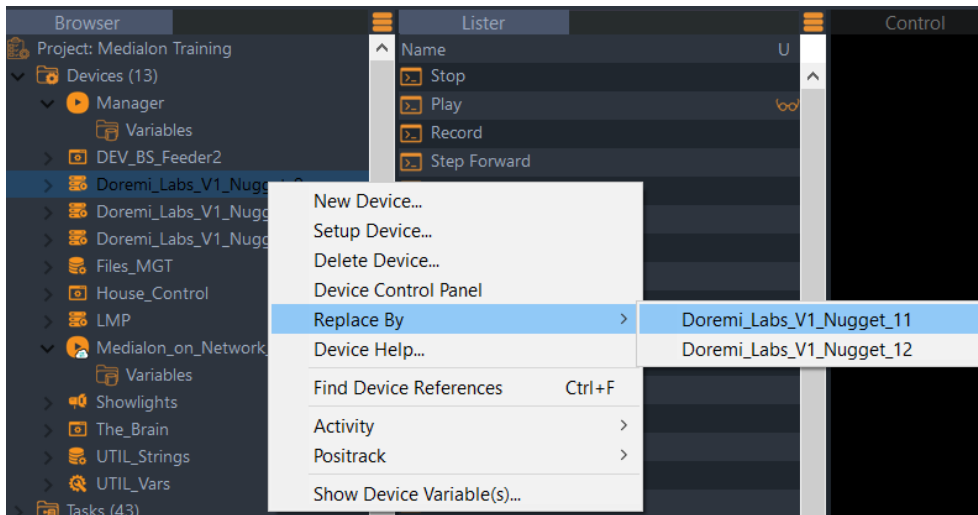
If a device command cue has been inserted into a task, it can be reassigned to a different device if the New device is identical (uses the same MxM). This allows you to keep your existing cues as they are, but send the commands to a different device.

To access the Cue Replace Device function, you must first select the new device the cue will be assigned to (in the Browser tab, the Lister tab or the Device Map). Once the target device is selected, right-click on the desired cue, and go to the 'Devices' sub-menu. Select 'Replace Device' to perform the reassignment.



Device Replacement

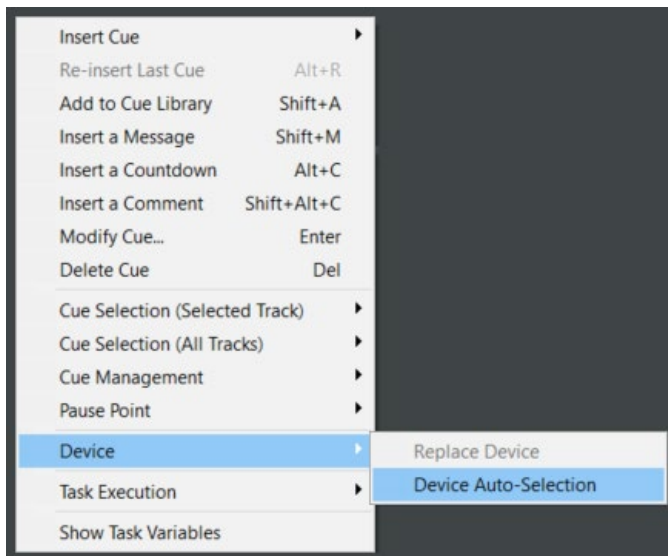
It is also possible to completely replace one Device by another Device, provided they are identical. All Device Command cues and variables that have been used in the project are reassigned to the target device. To completely replace one device by another, right-click on the device in the Browser tab, Lister tab or Device Map and select 'Replace By'. If there are no devices to replace it, the option is not available. If there are other devices in the project, only the compatible devices will be shown. Select the target device in the list to perform the replacement. All task cues and variables will be reassigned to the new device.



Device Replacement applies only to commands; it does not update tasks or cues that refer to device variables in their parameters or expressions. These need to be manually reprogrammed.

Device Auto-Selection

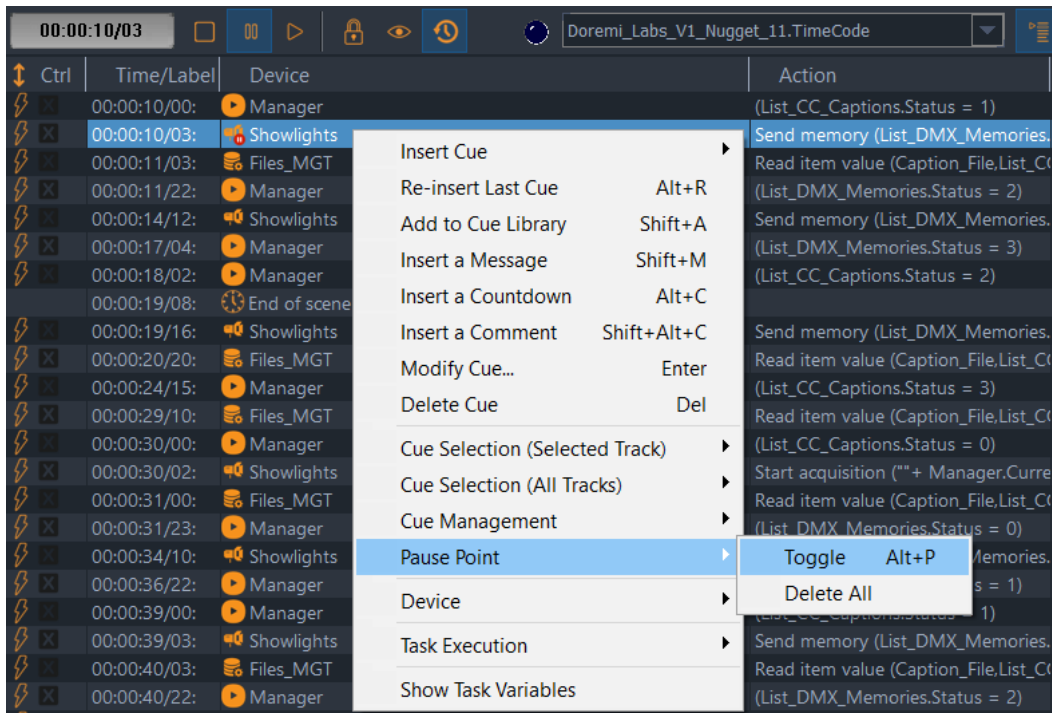
When this option is active, each time you select a cue in a task, the device assigned to the cue is selected. The next Insert Cue command will use the last device selected.



7.5.3 Pause Points

This function is not available in Run Mode and is handy for 'stepping through' while debugging complicated tasks.

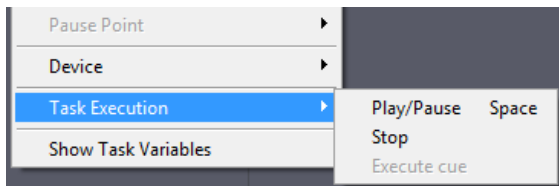
When a task is running and it reaches a Pause Point, the task pauses just before executing the cue and waits to be started again. Any existing cue in a task can be set as a Pause Point.



Select a cue and type Alt+P (or choose 'Pause Points/Toggle') to toggle a Pause Point. Choose 'Pause Points/Delete All' to remove all Pause Points from the task.

7.5.4 Task Execution

Instead of using task transport buttons or Flash controls, you can use the right-click menu to control the execution of a task.



Play/Pause Stop


Open a task in a task editor. In the right-click menu, select 'Task Execution/Play' to start the current task.

i The space bar can also be used to start or pause a task.

Execute Cue

From the right-click menu, select 'Execute Cue' in the 'Task Execution' section to perform the selected cue.

This command acts like a flash control.


-  This method can be used when Flash commands are not available (with Panel edition, for example).

7.6 Import and Export Project

7.6.1 Importing a Project file

Medialon Manager V7 was designed to allow projects to be imported into one another. This allows you to integrate previous projects into your current projects to minimize programming time and avoid repeat programming.

In the case of multiple imports, or if you import a project including names already used in the current project, (devices, variables, tasks, UserScreens), Medialon Manager V7 will rename the elements concerned by adding '_NEW' and an incremental number to them.

-  Manager V7 projects and tasks can only be imported into other Manager V7 projects. Showmaster V2 projects and tasks can only be imported into other Showmaster V2 projects. Showmaster V2 files cannot be imported into Manager V7 projects or vice-versa.

7.6.2 Exporting / Importing a Task

Exporting a task

Tasks may be exported to a file to be imported into another project. Each task you wish to export is stored in a separate file that cannot be opened by itself, only imported into a project.

To export a task, select the desired task in the Browser or the Lister. Then select 'Export...' in the File menu.

The 'Save Export...' dialog box appears. Choose a name and destination for the export file.

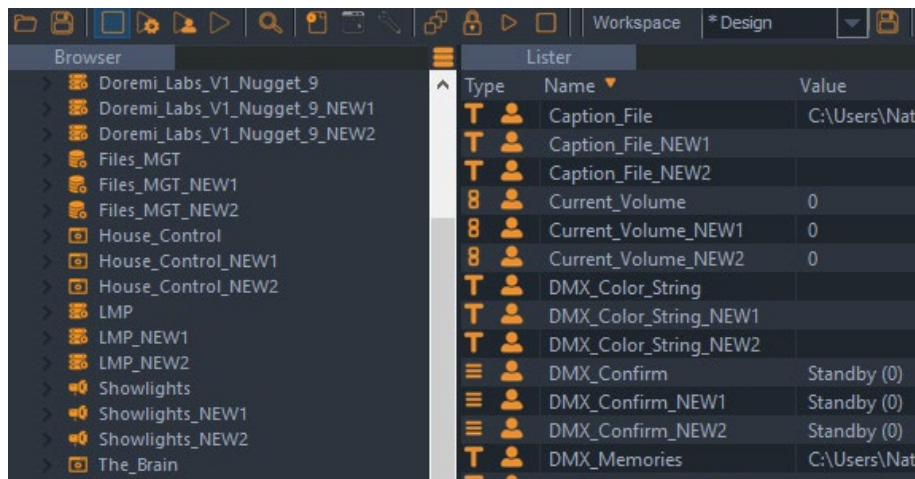
- taskname.mngExport for Medialon Manager V7 Export File

Importing a Task File

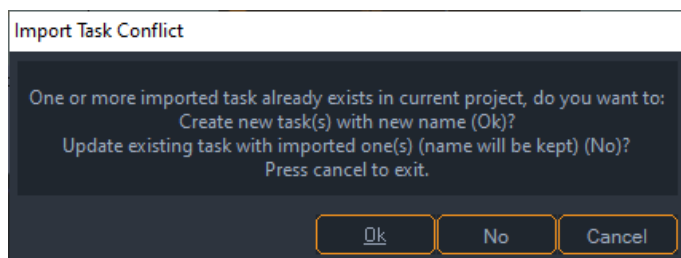
Tasks that have been exported from a project into a .mngExport file can be imported using 'Import' in the File menu.

The import file dialog box opens with the default file filter set to 'Medialon Export Files'. Select the desired export file to be imported and click 'Open'.

If the name of a *full show* export file is identical to one already in the project, the duplicate will be imported with the suffix `_NEW#`:



If you import *single tasks* duplicated in the current project, the 'Import Task Conflict' dialog box appears.



You may choose to import the task as a new task with a new name. The new name is automatically composed of [imported task's name]_NEW[#].

Alternatively, you may choose to update the existing task by the imported task and keep the same name. This cannot be undone and should only be chosen after having saved the project as a new version.



Exported tasks should only be imported into an identical project where the Device's names and types are identical.

7.6.3 Converting a Project

Medialon Manager projects can be converted into documents for analysis.

- XML document (*.xml)
- HTML document (*.html)
- Text document (*.txt)

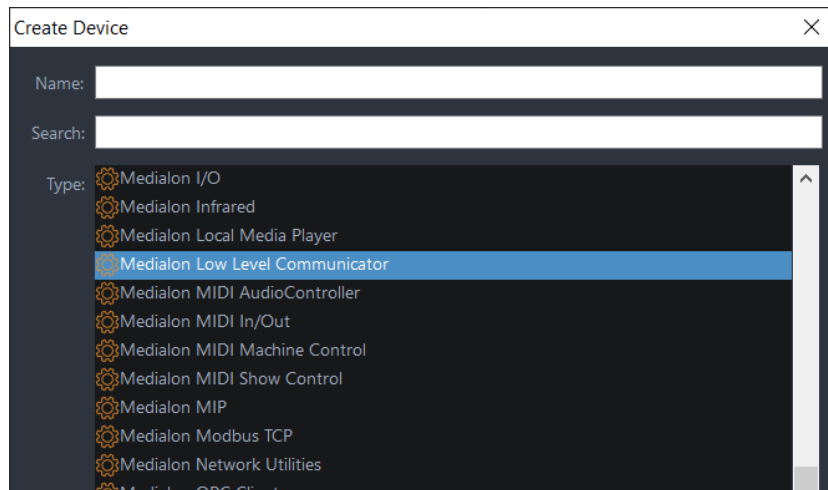
All the information of the project is converted into the selected format, but cannot be re-opened or converted back into a valid Project File. This is for documentation purposes only.

To convert a project, select 'Convert Project...' from the File menu. The 'Save Export' dialog box appears. Name the converted file.

7.7 Driver Editing

Even though there is a long list of available MxMs, you may need to control a device that does not have a dedicated MxM. Medialon Manager gives you a tool to create custom devices with commands and monitored variables. Your drivers can then be saved and reused in other projects.

To create a new MxM, right-click Devices > New Device (or Menu: Object > Devices > New Device) and select Medialon Low Level Communicator:



Each time you create a device using this plug-in, it will spawn its own individual device instance, which needs to be configured for the particular application, and saved as .mllc files.

Note: You can do the same to create new .js JavaScript scripts, by selecting 'Medialon Script'.

7.7.1 MxM Medialon Low Level Communicator

The MxM Medialon Low Level Communicator is a powerful low-level MxM not dedicated to any specific device and is able to use many raw communication protocols.

Medialon Manager programmers can write their own custom drivers for specific devices and projects.

The following section explains how create, edit, and save Low Level Communicator drivers. A full description of the MxM Medialon Low Level Communicator is available in the MxM help file.

7.7.2 Driver Creation

At the creation of the device, a setup dialog box appears.

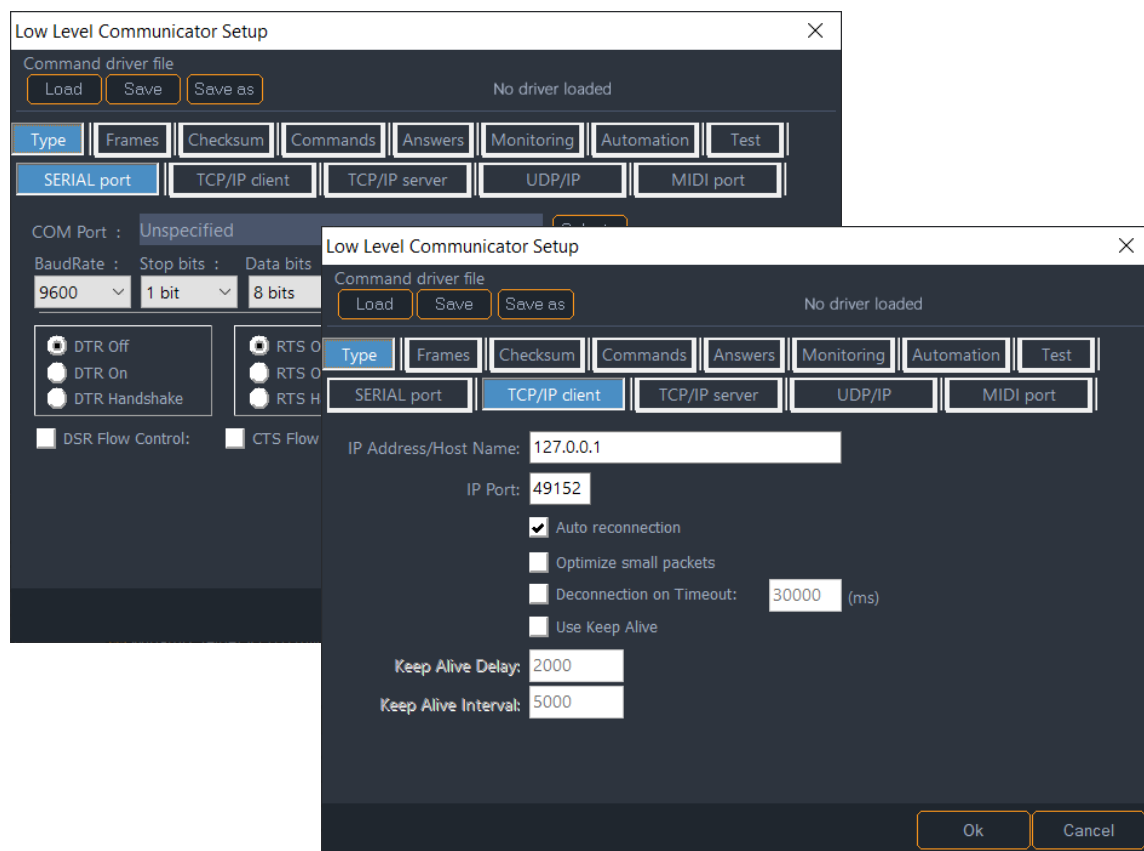
The first step of driver creation is to select the type of communication.

Type

Five types of communication are available:

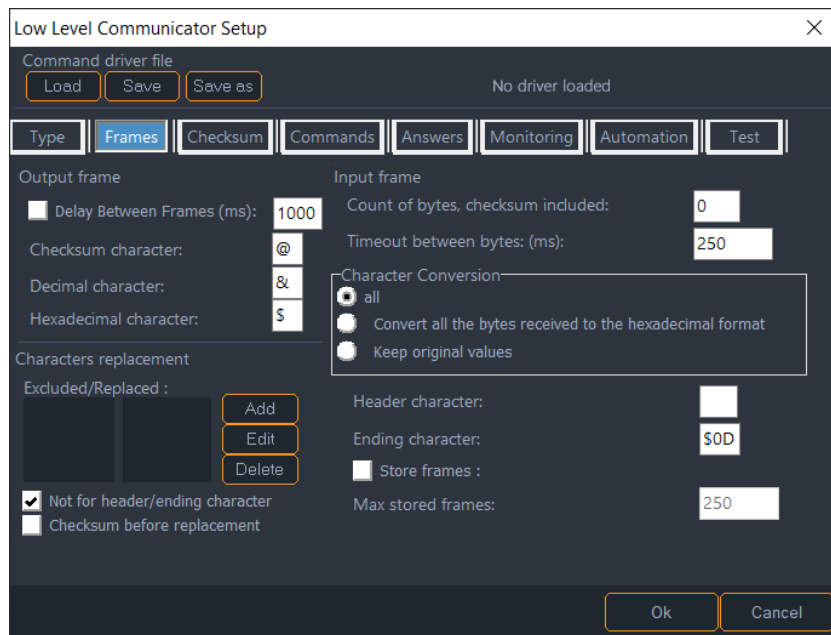
- Serial Port
- TCP/IP client
- TCP/IP server
- UDP/IP
- MIDI Port

Select and set up the required communication type and resource. Only one Type needs to be configured. The selected Type tab determines the communication.



Frames

The second step is to define the format for sent and received frames.



Output Frame

Characters used in the format of the output frames can be modified.

- **Checksum character:** a character that represents the checksum in an outgoing frame. It is computed before sending each frame. By default, the '@' character is used. Insert this character at the checksum position in the output frame. The checksum is calculated according to the settings from the Checksum tab and inserted in place of the '@' character.
- **Decimal character:** This is the ASCII character used for sending a decimal character. By default, '&' is used; write &255 for sending this upper ASCII byte (sends the bit string '11111111').
- **Hexadecimal character:** This is the ASCII character used for sending a hexadecimal character. By default, '!' is used. The two characters following the Hexadecimal character are evaluated to produce the output byte. The notation !FF sends the bit string '11111111'. Write !0D!0A for sending a carriage return (!0D) and a line feed (!0A).

Input Frame

These parameters define the alignment of incoming frames.

- **Header character:** When this character is received, it is considered to be the first character of a frame. You can use a hexadecimal byte here (for example !02).
- **Ending character:** When this character is received, it is considered to be the last character of a frame. You can use a hexadecimal byte here (for example !03).

Depending on the protocol and your needs, you can select the way the incoming frame is formatted when displayed in the 'LastFrame' variable of the device.

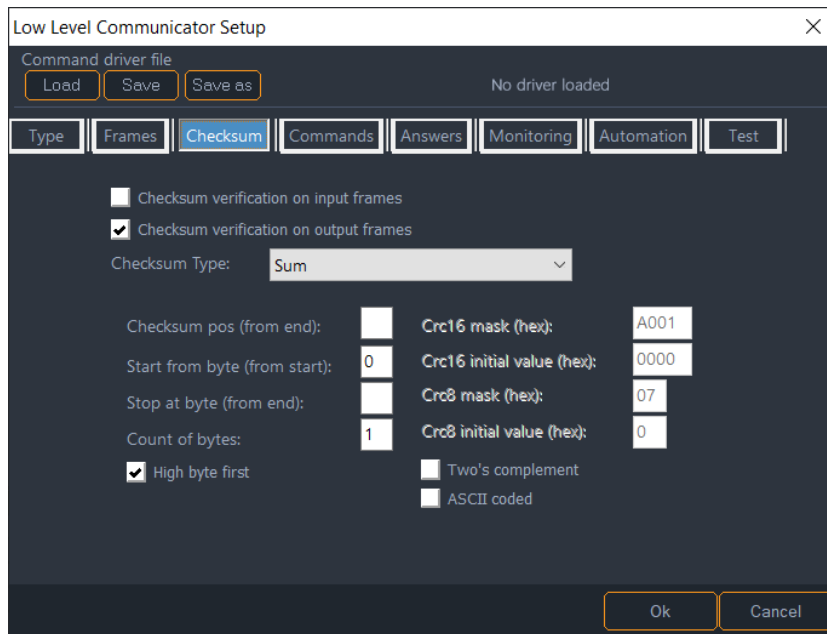
- Convert only the non-literal bytes to the hexadecimal format: If a non-literal byte (ASCII value <31 or >127) is received, it is converted to hexadecimal format, using the 'hexadecimal character' entered to the left.
- Convert all the bytes received to the hexadecimal format: All bytes received are converted to hexadecimal format, using the 'hexadecimal character' entered in the box to the left.
- Keep the original values: When checked, nothing is converted and all non-printable characters appear as strange characters, spaces, or little boxes.

Characters Replacement

This section lists the characters that are replaced in frames before sending and replaced back the other way upon reception. The left list contains the excluded characters, the right list contains their replacement. On output, excluded characters are replaced by their corresponding replacement string, and on input, the replacement string is transformed back to the excluded characters.

Checksum

These parameters define the checksum verification options, if needed. They work together with the 'checksum character' and 'checksum before replacement' options of the 'Frames' config tab.



Checksum verification can be enabled on input or output frames, or on both. The first occurrence of the 'checksum character' is replaced when sending or receiving by the appropriate checksum, computed using the following options:

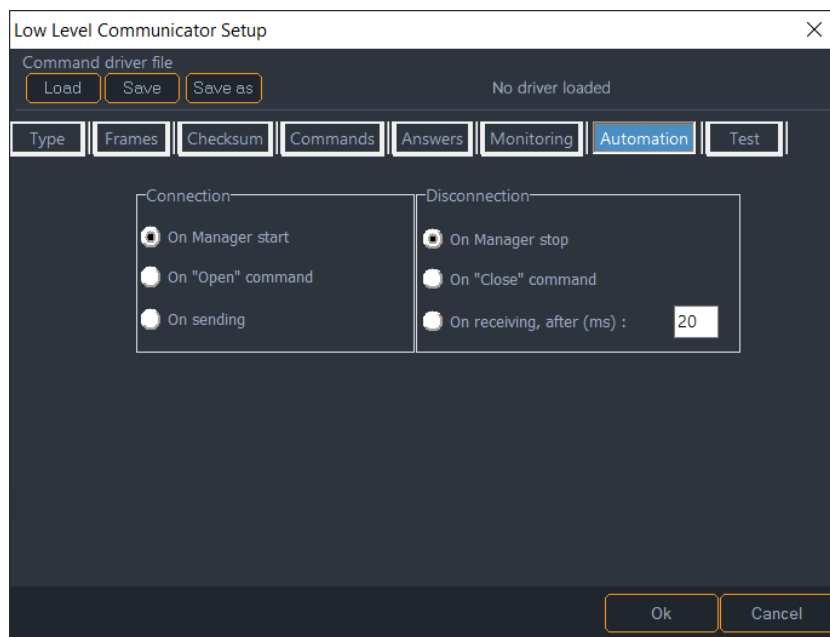
Checksum Type:

- *Sum*: The checksum is the sum byte of the bytes between the byte indicated for start and the byte before the checksum character.

- *Xor*: The checksum is the byte resulting of an XOR operation between the byte indicated for start and the byte before the checksum character.
- *Or*: The checksum is the byte resulting of an OR operation between the byte indicated for start and the byte before the checksum character. This is a very inefficient checksum but it is present in some protocols.
- *Crc16*: The checksum is the two bytes resulting of a polynomial calculation between the byte indicated for start and the byte before the checksum character (Crc = cyclic redundancy check).
- *Level Control System 7 Bit Checksum*: This type of checksum is used by the Level Control System (LCS) Matrix3 protocol.
- *Alcorn McBride AMINet*: This type of checksum is used by the Alcorn McBride AMINet UDP protocol (for the DVM4).

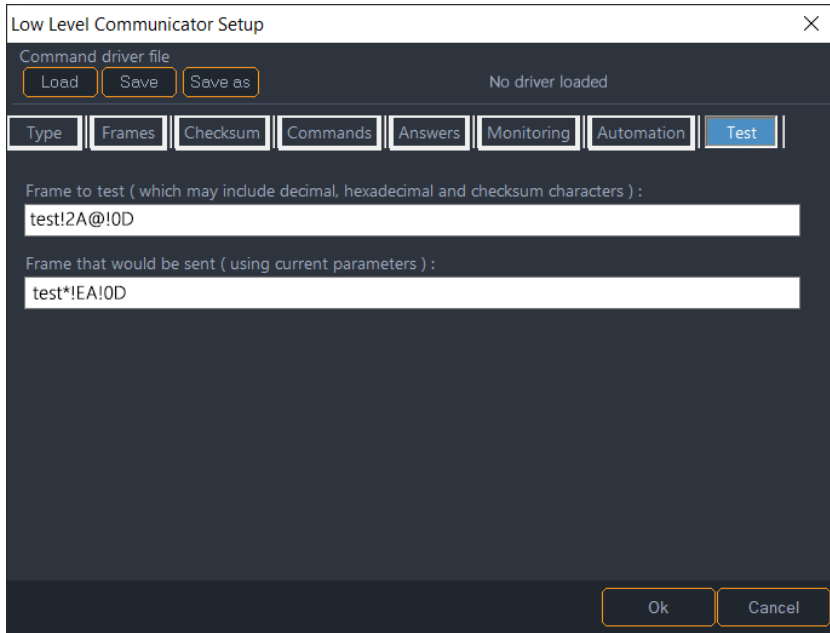
Automation

In the automation page, you can define the device connection / disconnection behavior.



Test

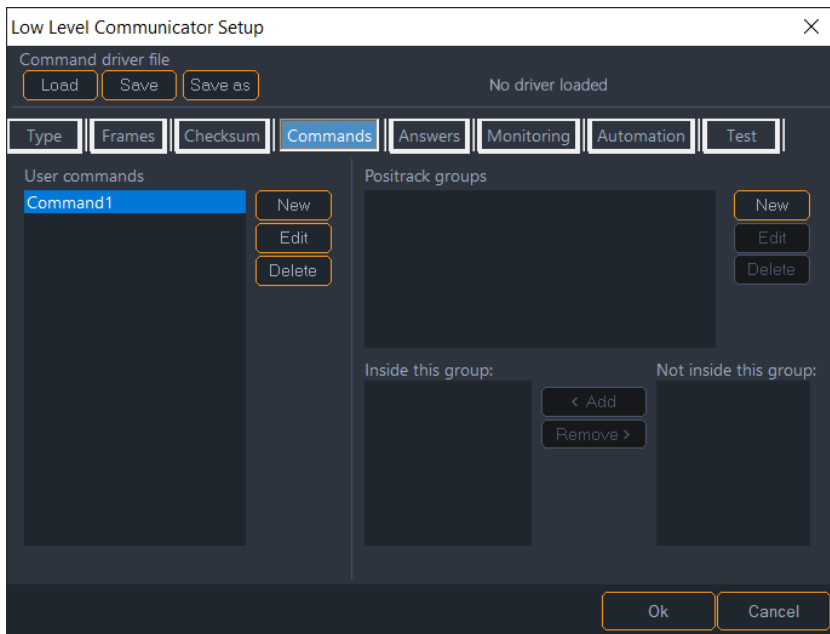
With the Test page, you can check all your frame settings, such as checksum or formatting. In the top box, type in a frame with ASCII, Hexadecimal, Decimal, Checksum, and replacement characters and see the result of what will be sent in the bottom box.



7.7.3 User Commands

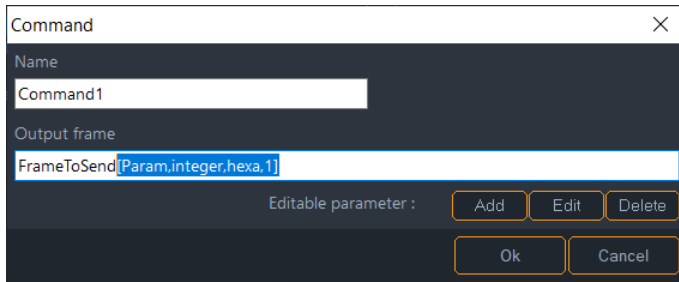
Once you have defined the communication type and frame format according to the protocol, you can add commands to the device.

Click on the 'New' button to create a new user command or select a command in the list and click the 'Edit' button to modify an existing command.



Command Creation

When you create a new command, specify the name and the output frame of the command.



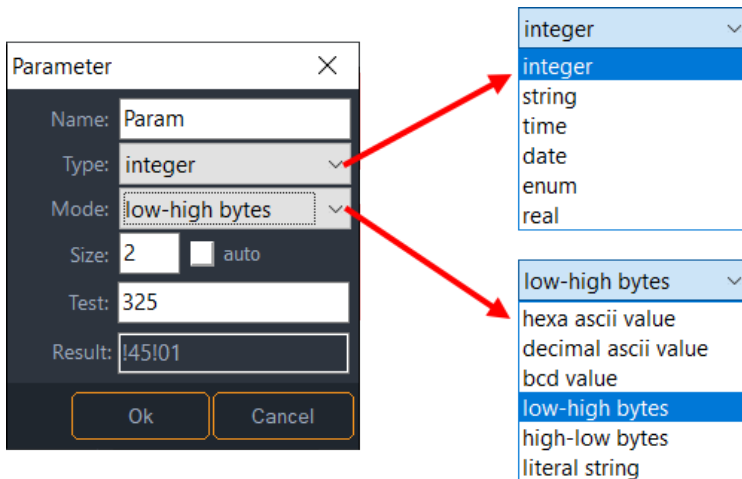
It is important to include any header or ending characters required by the protocol, even though they have been specified in the Frames tab. The settings in the Frames tab only tell the Low Level Communicator MxM how to define the edges of the Frames. Every command must include all bytes to be sent to the device.

In the output frame, it is possible to define parameters that will appear as a Property of the Device's command. For instance, a Power command can use a parameter to set the power 'On' or 'Off'.

To add an editable parameter to your command, click the Add button to open the Parameter dialog box. The parameter's result is inserted into the outgoing frame.

Command Parameters

In the output frame, the syntax of the parameters is between brackets. It contains the name, the type, and the size of the parameter.



Five types of parameters can be specified:

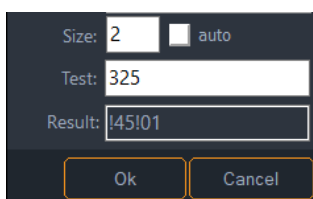
- Integer
- String

- Time
- Date
- Enum

Select the mode of conversion needed to fit with the output frame format. Six modes can be used:

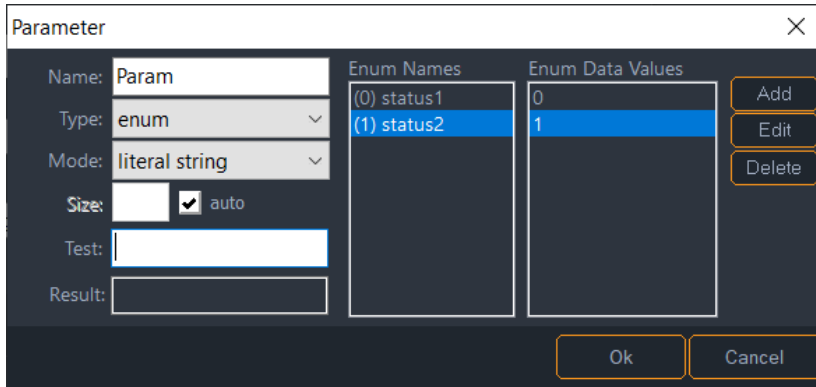
- *hexa ascii value*: the resulting bytes in the frame will be the ASCII representation of the hexadecimal value of the parameter. The value of the parameter is first converted in a hexadecimal integer, then this integer is converted in ASCII.
Example: Integer value 325 converted as !145 will generate the bytes !31 !34 !35 (ASCII code '145').
- *decimal ascii value*: the resulting bytes in the frame will be the ASCII representation of the decimal value of the parameter. The value of the parameter is first converted in a decimal integer, then this integer is converted in ASCII.
Example: Integer value 325 will generate the bytes !33 !32 !35 (ASCII code '325').
- *bcd value*: the resulting bytes in the frame will be the binary-coded-decimal value of the parameter. The value of the parameter is first converted into a decimal integer, then each power of 10 of this value is converted into bytes.
Example: Integer value 325 will generate the bytes !03 and !25.
- *low-high bytes*: the resulting bytes in the frame will be the binary value of the parameter, with the low byte first. The value of the parameter is first converted in a hexadecimal integer, then the resulting bytes are organized low bytes first.
Example: Integer value 325 converted as !145 will generate the bytes !45 and !01.
- *high-low bytes*: the resulting bytes in the frame will be the binary value of the parameter, with the high byte first. The value of the parameter is first converted in a hexadecimal integer, then the resulting bytes are organized high bytes first.
Example: Integer value 325 converted as !145 will generate the bytes !01 and !45.
- *literal string*: the resulting bytes in the frame will be the string value of the parameter. If the parameter is a string or an enum, the string value is keep as it is, otherwise it is converted into an integer and then expressed into a string.
Example: Integer value 325 will generate the bytes !33 !32 !35 (ASCII code '325').

Test and Result fields can be used to check the selected conversion mode



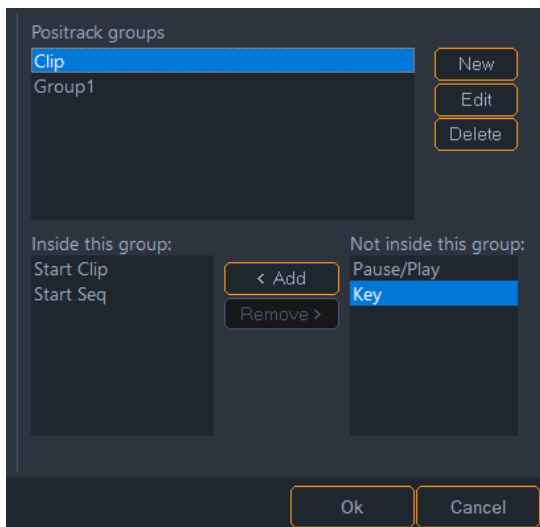
The size represents the minimum count of bytes that is sent: if the result of the conversion is smaller, the number is padded with zeros to the left.

The enum type is different than the others because for each item of the enum, you must specify the name and the value to be sent:



Positrack Groups

Each command can belong to Positrack groups. If several commands belonging to the same Positrack group are used in a timeline that is put in pause mode, only the last command before the pause point is executed.

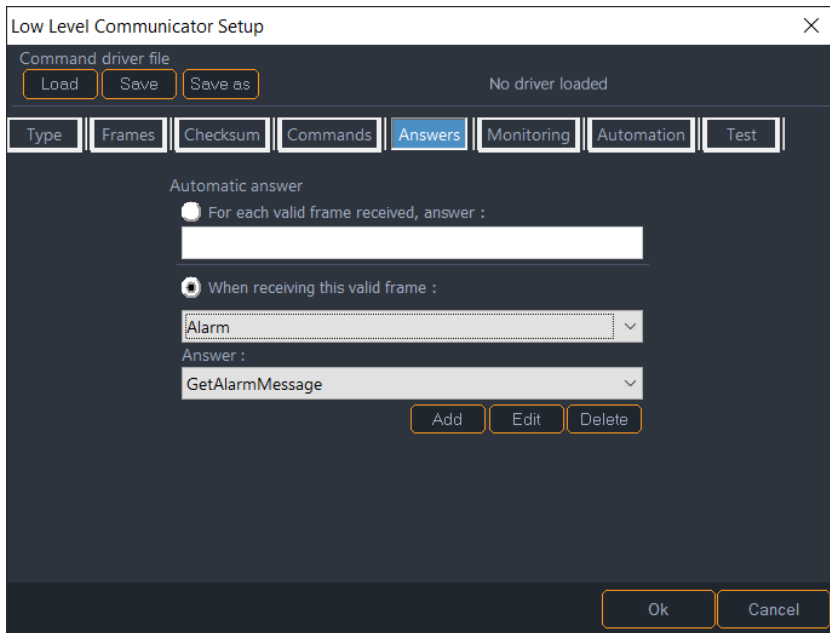


7.7.4 Monitoring, Request and Variables

Medialon Manager can send periodic polling requests and automatically update device variables from the monitored responses.

Automatic Answers

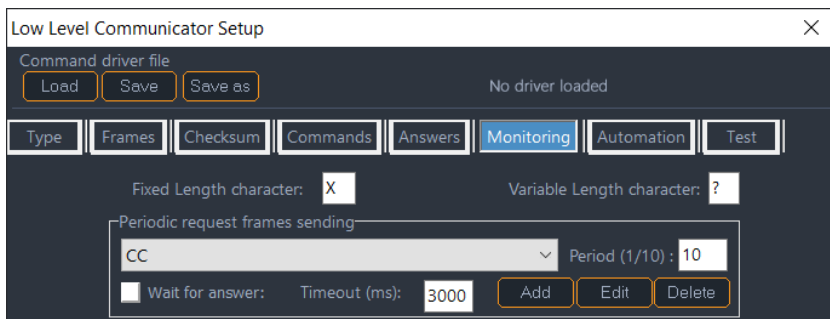
Some protocols require specific answers before sending the next frame; this section defines these automatic answers.



Periodic Request

The Periodic Request feature repeatedly sends frames directly from the device without the need of creating a task to do it. This is handy for devices that must be ‘polled’, or asked for information.

Click on ‘Add’ to create a new polling request.



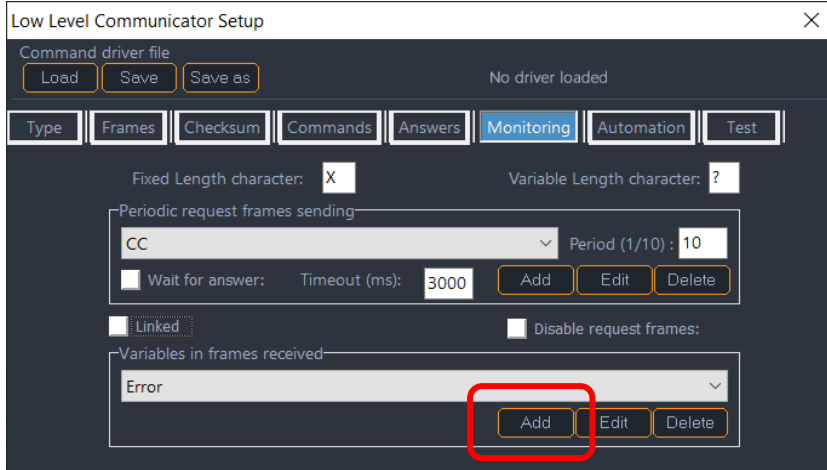
The period is the time (in tenths of a second) between which the selected request is sent. Each request can have a different polling period.

If ‘Wait for answer’ is checked, the device will wait for an answer before another request is sent. If the answer is not received within the time specified in the ‘Timeout’ parameter, the next request is sent.

While you are testing your driver, you may need to disable the periodic sending of the request frames to prevent polling requests from interfering with tests you may be conducting. The ‘Disable request frames’ option allows you to turn off all automatic polling.

Variables

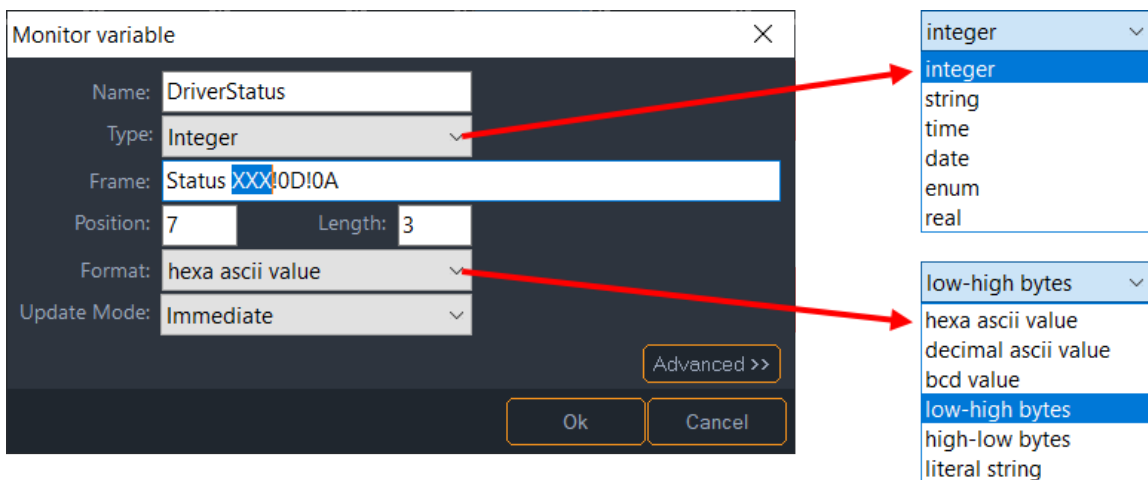
From incoming frames, Medialon Manager can extract data and set the value of a device variable. Click the 'Add' button in the 'Variables in frames received' section to create a new device variable.



Incoming frames contain patterns; some bytes are 'static' and other bytes can change. This changing pattern is defined with wildcard characters:

- Fixed length character: a wildcard character that represents one unknown byte (default is 'X').
- Variable length character: a wildcard character that represents an unknown variable length character string (default is '?').

After entering the name and the type of the variable, enter the frame, containing both static bytes and unknown bytes. To determinate the position and the length of the bytes that will be monitored by the variable, highlight these bytes in the frame before closing the dialog box. Multiple unknown bytes are allowed in a frame. The highlighted bytes are the ones that will update the variable.



Five types of variables can be specified:

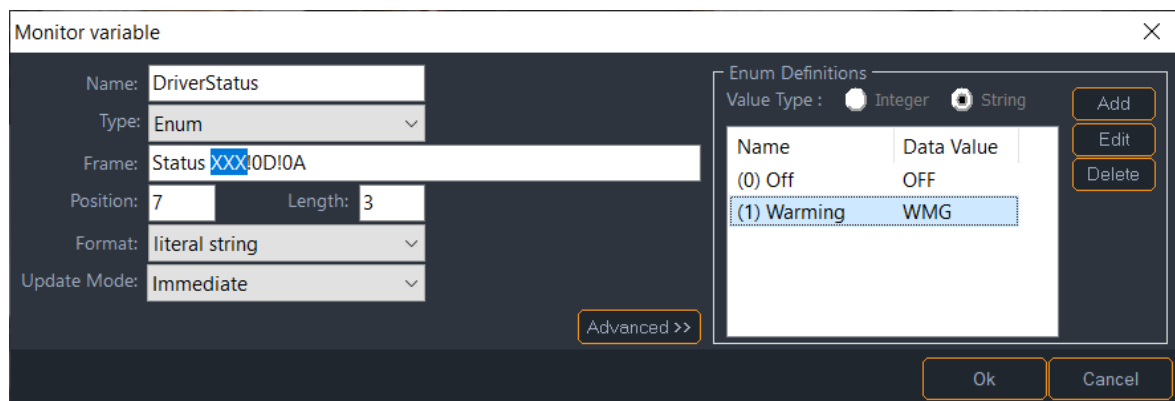
Advanced Programming

- Integer
- String
- Time
- Date
- Enum

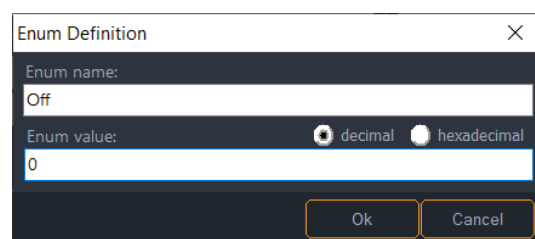
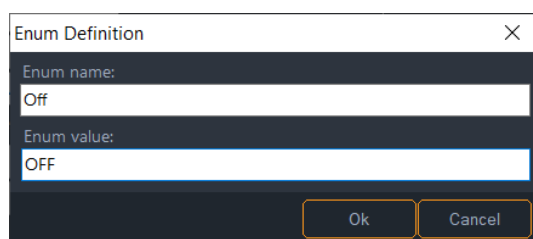
Select the input frame format; six formats can be used:

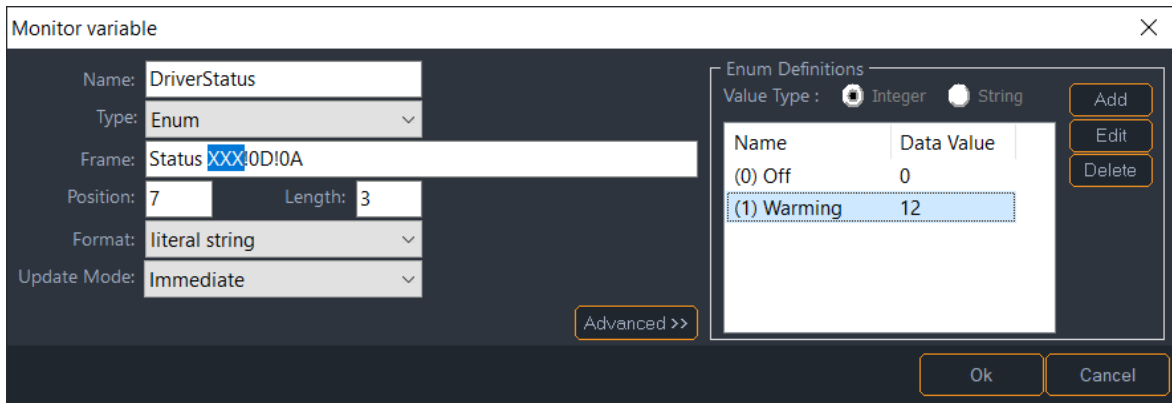
- *hexa ascii value*: the bytes are the ASCII representation of the hexadecimal value of the parameter. ('9' returns '9', '255' returns 'FF')
- *decimal ascii value*: the bytes will be the ASCII representation of the decimal value of the parameter. ('9' returns '9', '255' returns '255')
- *bcd value*: the bytes will be the ascii representation of the binary-coded decimal value of the parameter. ('9' returns '!09', '255' returns '!02U')
- *low-high bytes*: the bytes will be the binary value of the parameter, with the low byte first. ('9' returns '!09', '500' returns '!F4!01')
- *high-low bytes*: the bytes will be the binary value of the parameter, with the high byte first. ('9' returns '!09', '500' returns '!01!F4')
- *literal string*: the bytes will be the string value of the parameter. This is the most common Mode. ('9' returns '9', '500' returns '500')

The enum type is different than the others because for each item of the enum, you must specify the name and the value to be received.

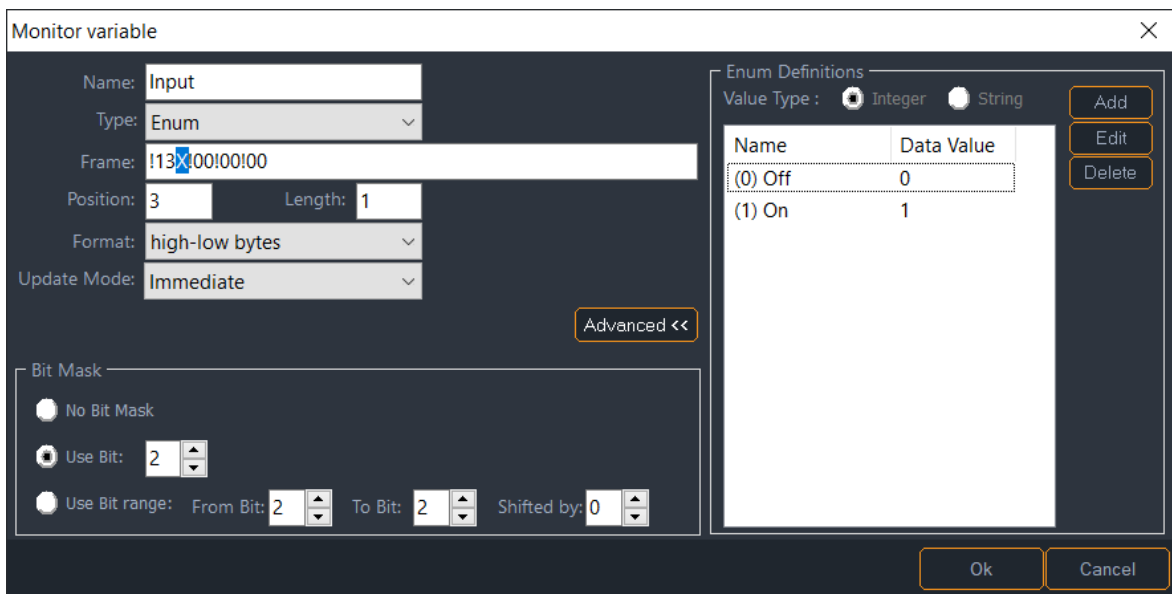


Incoming values can be Strings or Integers.

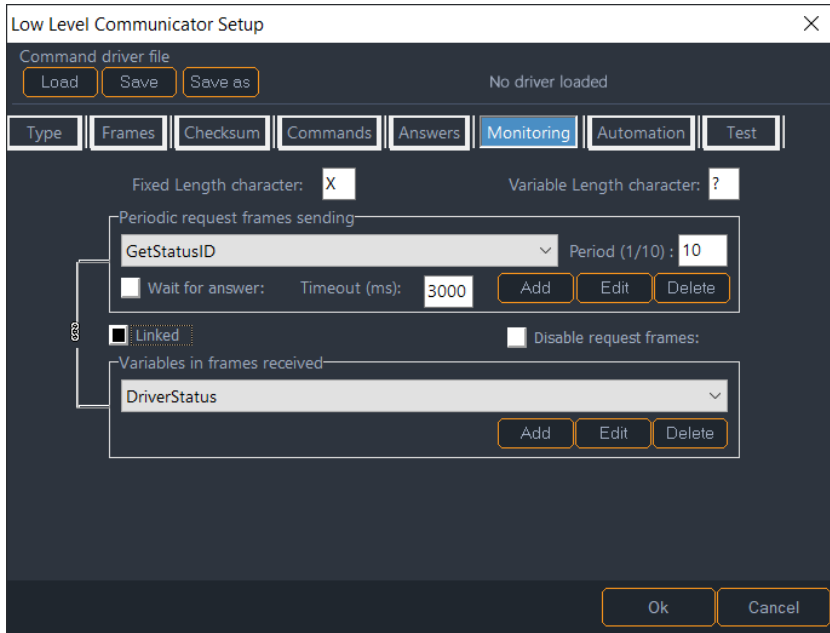




When the monitoring value embeds information in particular bits. A Bit Mask can be applied to integer or integer enum values.



With some protocols, different requests send back data with the same incoming frame syntax. It is possible to link a request and a variable in order to tie the received data to the request for that data. Check the 'Linked' box to tie a specific request to a specific variable.

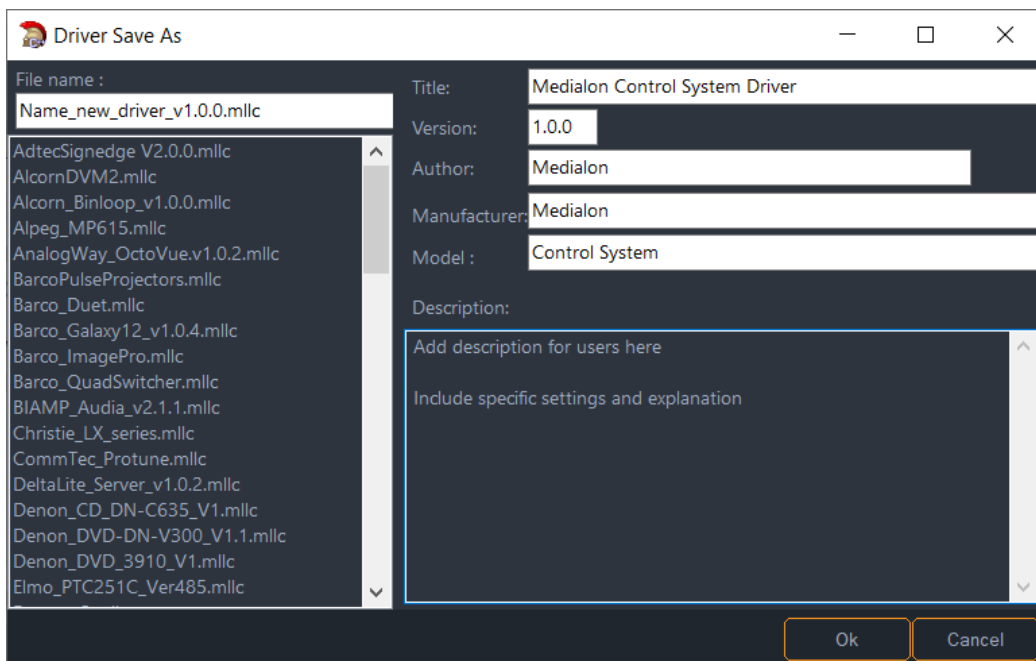


7.7.5 Save and Share Drivers

After creating and testing of your driver, you can save it and share it with other Medialon Manager projects or programmers.

Save / Save as

Click the 'Save' or 'Save as' button to save the current settings of the Low Level Communicator device as a driver file. The file is saved in the Low Level Communicator drivers directory inside the MxM directory.

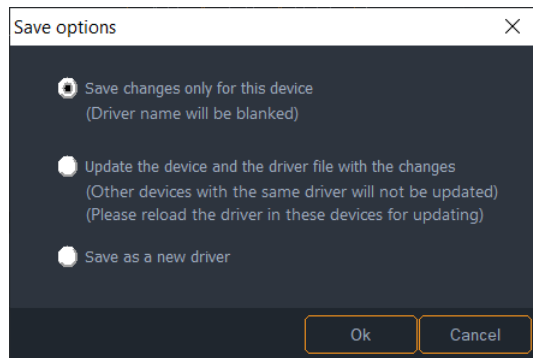


Fill in the info on the right side of the dialog box and remember to name the MLLC file.

Load

After the setup of the device is done, the driver file is not used since the Device's parameters are saved in the Project file.

If a driver has been previously loaded from a file and the parameters of this driver are modified, a dialog box asks how you want to save the modifications.



- Save Changes only for this device: the device's modifications will be saved in the project file only and will not affect the .mllc file.
- Update the device and the driver file with the changes: both the project and the .mllc file will be updated with the modifications to the device driver. However, the changes are not automatically applied to other devices of the Project that use the same driver. In order to apply the modifications, the driver file must be manually loaded in each of the devices.
- Save as a new driver: opens the Driver Save As... dialog box and allows you to save your driver as a new .mllc file.

Share Your Drivers

You have worked hard to create your device driver. Why not share it with other programmers by posting your driver on our website?

You can download drivers created by other Medialon Manager users. The website contains MxMs, MRCs, Low Level Communicator drivers, and MxM Medialon Script drivers, and is always growing as programmers submit their own drivers for new devices.

To submit a driver, email it to support@7thsense.one.

7.8 User Mode

Medialon Manager's User Mode allows the programmer to decide elements are accessible to the user (which tasks, which devices, which commands for devices, etc.). User Mode works like the Debug Mode, but only with selected objects.

User Mode  is only accessible from Stop Mode by typing *Ctrl+F2* or by clicking on the User Mode button.

7.8.1 User Mode Rights

The User Mode Rights property defines the behavior of objects when Medialon Manager V7 is in User Mode.

In User Mode, an object with no rights is not listed in the Browser or Lister; the user cannot select or monitor it. However, all programmed commands sent to this object are executed normally, like in Run Mode.

Objects with View rights are listed in the Browser and Lister; the user can select or monitor them. An object with View rights can be used and edited in tasks if the task has Edit rights.

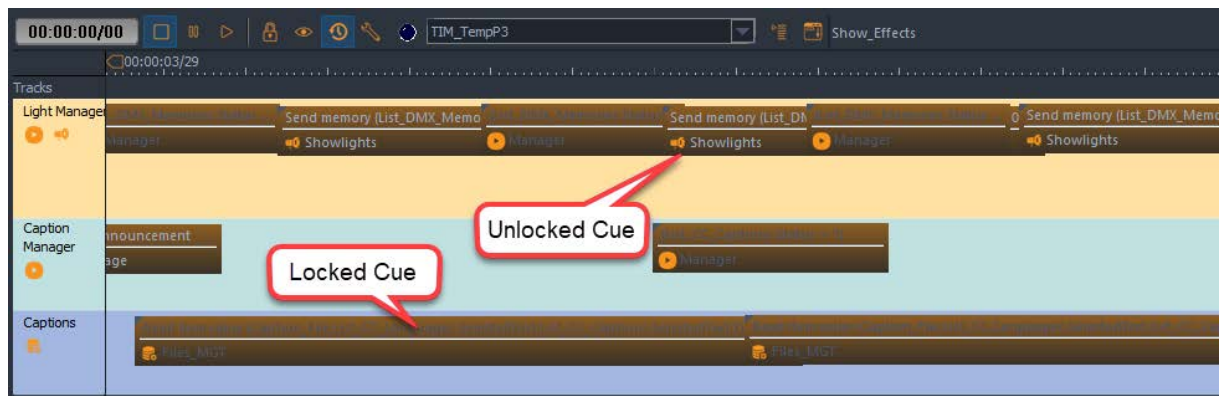
User Mode Rights and Object interactions:

Rights	None	View	Edit
Device	Device is not visible in the browser. Cues (in tasks) cannot be edited.	Device is visible in the browser and lister. Cues (in tasks) can be edited.	N/A
Panels	Panel is not visible in the browser or lister.	Panel is visible in the browser and lister.	N/A
UserScreen	UserScreen is not visible in the browser or lister.	UserScreen is visible in the browser and lister.	N/A
Variables	Variable is not visible in the browser or lister.	Variable is visible in the browser and lister.	N/A
Cue Library	Cue Library is not visible in the browser or lister.	Cue Library is visible in the browser and lister.	N/A
User Group	User Group is not visible in the browser or lister.	User Group is visible in the browser and lister.	N/A
Task	Task is not visible in the browser.	Task is visible in the browser and lister. task can be opened in a Task Editor (read only)*. Task Control Panel can be opened.	Task is visible in the browser and lister. task can be opened and edited in a Task Editor like in Debug Mode.

(*) Option available in Tasks Preferences.

7.8.2 Task Editor in User Mode

When you open a task with Edit user rights in User Mode, editing cues is possible if the affected Devices and Commands have 'View' user rights.



If a Device's rights are set to 'None', all its cues are locked (dark blue text) and you are not allowed to change, move, or delete the cues. Unlocked cues have text in light blue, and are not greyed out in the list view:

The screenshot shows a list of cues in the Medialon Manager V7 interface. The list has columns for Ctrl, Time/Label, Device, Action, and Comment. The cues are as follows:

Ctrl	Time/Label	Device	Action	Comment
	00:00:00/00:	House_Contr	Houselights (0)	
	00:00:01/26:	Manager	(List_CC_Captions.Status = 0)	
	00:00:02/53:		verbal announcement	
	00:00:02/94:	Manager	(List_DMX_Memories.Status = 0)	
	00:00:03/66:	Files_MGT	Read item value (Caption_File,List_	
	00:00:05/41:	Showlights	Send memory (List_DMX_Memories	
	00:00:07/90:	Manager	(List_DMX_Memories.Status = 1)	
	00:00:10/00:	Manager	(List_CC_Captions.Status = 1)	
	00:00:10/13:	Showlights	Send memory (List_DMX_Memories	
	00:00:11/13:	Files_MGT	Read item value (Caption_File,List_	
	00:00:11/88:	Manager	(List_DMX_Memories.Status = 2)	
	00:00:14/48:	Showlights	Send memory (List_DMX_Memories	
	00:00:17/17:	Manager	(List_DMX_Memories.Status = 3)	
	00:00:18/09:	Manager	(List_CC_Captions.Status = 2)	
	00:00:19/34:		End of scene	
	00:00:19/65:	Showlights	Send memory (List_DMX_Memories	
	00:00:20/81:	Files_MGT	Read item value (Caption_File,List_	

If Device's rights are set to 'View', cues can be modified, just like in Debug Mode.

8 Menus and Wizards

This section describes your Medialon Manager software's menus and other wizards.

8.1 Menus

8.1.1 File Menu

New Project (Ctrl+N*)

Creates a new project.

Open Project (Ctrl+O*)

Opens a saved project.

Open Recent Project...

Shows the last recently used files. Up to 10 files are listed.

Convert Project...

Converts the current project to a different file format.

Save Project (Ctrl+S*)

Saves the current project under the current name in the current directory.

Save Project as...

Saves the current project under another name and/or in another directory.

Import...

Merges a current project with all or part of another project. You can thus include UserScreens, variables and tasks already created in previous projects.

In the case of multiple imports, or if you import a project including names already used in the current project, (devices, variables, tasks), Medialon Manager will rename the elements concerned by appending '_NEW' and a number to the end of the elements' names. See also [Importing a Project file](#).

Export

Opens the Export dialog box.

Stop (Esc*)

Stops the project.

Debug (F2*)

Runs a project with the workspace windows still open. This enables you to visualize and test the project, while still being able to adjust variables and tasks for debugging purposes.

User (Ctrl+F2*)

Runs a project with the workspace windows still open, displaying a limited subset of devices, tasks, UserScreens, and variables. This enables you to visualize and test selected objects in your project.

Run (F3*)

Runs the project in its final user format (no workspace interventions possible).

Printer Setup

Opens the standard Windows Printer Configuration dialog.

Print (Ctrl+P*)

If the active tab is a Variable Lister tab, it prints the list of the variables in the Lister. If the active tab is the Task Lister tab, it prints the list of the tasks in the project. Otherwise, the content of the active Area is printed.

Preferences...

Opens the 'Preferences' window for configuring Medialon Manager V7. See also [Preferences](#).

Install a License File

Opens the 'License' browser to install a license to increase the number of panel connections. Contact your Medialon dealer for further information.

Quit (Ctrl+Q*)

Quits Medialon Manager V7. A save request displays if the current project has been modified since it was last saved.

8.1.2 Edit Menu

The actions triggered by the items in the Edit menu depend upon the currently active element in the active window. If an action is not allowed, the corresponding menu item is disabled.



Note that most of the menu items at the bottom of this menu are only applicable to graphic objects in a UserScreen.

Undo (Ctrl+Z*)

Undoes the last modifications. Changes to Graphical Objects in UserScreens cannot be undone.

Redo (Shift+Ctrl+Z)

Re-applies the modifications that were undone by the Undo command.

Cut (Ctrl+X)

Cuts the selected element to the clipboard.

Copy (Ctrl+C)

Copies the selected element to the clipboard.

Paste (Ctrl+V)

Pastes the content of the clipboard.

Select All (Ctrl+A)

Selects all elements in the active window. This command is available for:

- Cues in a Task Editor
- Objects in the Lister
- Graphic elements in a UserScreen

Group (Ctrl+G)

Groups the selected objects.



This function is now available by selecting UserScreen objects and right-clicking.

UnGroup (Ctrl+U)

Ungroups the selected group.

Find... (Shift+Ctrl+F)

Opens the Find dialog box to search for elements in the Project. See also [The Finder](#). The following commands are dedicated to UserScreens.

Lock (Ctrl+Y)

Locks the selected Graphic Objects. Locked graphical objects cannot be moved.



This function is now available by selecting UserScreen objects and right-clicking.

Unlock (Shift+Ctrl+Y)

Unlocks the selected Graphic Objects.

Send To Front

Brings the selected Graphic Objects to the foreground.

Send To Back

Pushes the selected Graphic Objects to the background.

Pull To Front (Ctrl+F)

Pulls the selected Graphic Objects one level towards foreground.

Push To Back (Ctrl+B)

Pushes the selected Graphic Objects one level towards background.

Align... (Ctrl+L)

Opens the Align dialog box for aligning selected Graphic Objects.

Magnetic grid

Turns the UserScreen Magnetic Grid On/Off.

In the 'UserScreens', the Magnetic Grid enables you to position objects in relation to each other at distances defined by the virtual grid. This grid is shown by points on the 'UserScreens'. When the Magnetic Grid is turned off, objects can be moved freely without snapping to grid points.

8.1.3 Object Menu

Edit...

Opens the Editor of the selected task object.

Duplicate (Ctrl+D)

Duplicates the selected object.

Delete

Deletes the selected object (cannot be undone).

Setup

Opens the setup window of the selected object.

Control Panel

If the selected object has a Control Panel (time based tasks or Devices), the Control Panel opens in the Control Tab.

Panels

New WebPanel...

Create a new WebPanel.

UserScreens

The items of this submenu apply only to the UserScreens. The UserScreens are graphic interfaces that enable the end user to interact with the application. UserScreens are the visible parts of the programming. They handle user requests, trigger tasks, and report back statuses from the devices and variables in the project.



Many commands in this menu are also available in the Panels Editor toolbar or by right-clicking in the Panels Editor.

New UserScreen

Creates a new UserScreen in the current Panel.

Goto Previous Page

Changes the current page of the active UserScreen to the previous one.

Goto Next Page

Changes the current page of the active UserScreen to the next one or creates a new page if the current page is the last configured page of the UserScreen.

Goto First Page

Changes the current page of the active UserScreen to the first one.

Goto Last Page

Changes the current page of the active UserScreen to the last one.

Move Objects

Enables you to move one or several selected objects from one page to another. This command works not only in the current UserScreen but also between various UserScreens in the project. Moving objects, as opposed to cutting and pasting them, leaves all associated programming intact.

Devices

The items in this submenu provide for the manipulation of Devices.

New Device... (Ctrl+Ins)

Creates a new device.

Replace by...

Replaces the selected device by a compatible device currently in the project. This option is only available if a compatible device is present.

Activity...

Activates or deactivates the selected device. The commands associated with the deactivated Devices are not executed. Variables of a deactivated device may or may not reflect the actual state of the Device. When a Device is deactivated, a red background is drawn on it to remind you of its state.

Positrack...

Activates or deactivates Positrack.

Positrack is a mode that causes the selected device to automatically jump to what it should be doing at a given position of a time based task when the timeline jumps, provided the device supports that functionality. When a Device's Positrack mode is disabled, an icon (a red chain link) is drawn on it in the Device Map to remind you of its state.



Note that the 'Manager/Showmaster V2' device itself cannot be deactivated or have its Positrack mode set.

Show Device Variables

Displays variables from the selected device in the Lister.

Variables

New Variable... (Shift+Ins*)

Creates a new user variable. The type of the variable to create must be chosen.

Delete Unused Variables...

This command shows all unused user variables and allows the user to delete them. When the variable list dialog box appears, unused variables can be selected, either individually or by group, for deletion.

Tasks

New Task (Ctrl+T)

Creates a new task.

New Task Editor (Shift+Ctrl+T)

Opens an empty Task Editor window.

8.1.4 Windows Menu

Workspace

Save as Template

Saves the current Workspace configuration as a template, which can be recalled at any time.

Rename

Renames the currently selected custom workspace.

Delete

Deletes the currently selected custom workspace.

Tools

Timecode Calculator (Ctrl+F11)

Opens the Timecode Calculator.

The timecode calculator provides basic operators like + (add), - (subtract), * (multiply), and / (divide).

It also provides a timecode units selector (24 SMPTE fps, 25 EBU fps, 30 and 30DP NTSC fps, 100 fps, 1000 fps) and a timecode format selector (timecode or frames). Changing these settings automatically converts the content of the 'Input' and 'Result' fields, according to the frame rate selected.

This tool works as a stack, much like Reverse Polish Notation; input a value and click on the desired operator to get the result.

Image selector

Opens the Image Selector, See also [Image Selector](#).

View

Browser (F5)

Sets the focus on the Browser Tab or switches to it if the Tab is hidden by another Tab in the Area.

Lister (F6)

Sets the focus on the Lister Tab or switches to it if the Tab is hidden by another Tab in the Area.

Properties (F7)

Sets the focus on the Properties Tab or switches to it if the Tab is hidden by another Tab in the Area.

Device Map (F8)

Sets the focus on the Device Map Tab or switches to it if the Tab is hidden by another Tab in the Area.

Panels (F9)

Sets the focus on the Panels Tab, or switches to it if the Tab is hidden by another Tab in the Area.

Control Center (F10)

Sets the focus on the Control Tab, or switches to it if the Tab is hidden by another Tab in the Area.

Panels List (Ctrl+F5)

Opens the Panels item in the Browser Tab and displays the list of available Panels in the Lister Tab.

Device List (Ctrl+F6)

Opens the Devices item in the Browser Tab and displays the list of the project's Devices in the Lister Tab.

Variables List (Ctrl+F7)

Opens the user variables item in the Browser Tab and displays the list of these variables in the Lister Tab.

Task List (Ctrl+F8)

Opens the tasks item in the Browser Tab and displays the tasks of the project in the Lister Tab.

Close

All UserScreens

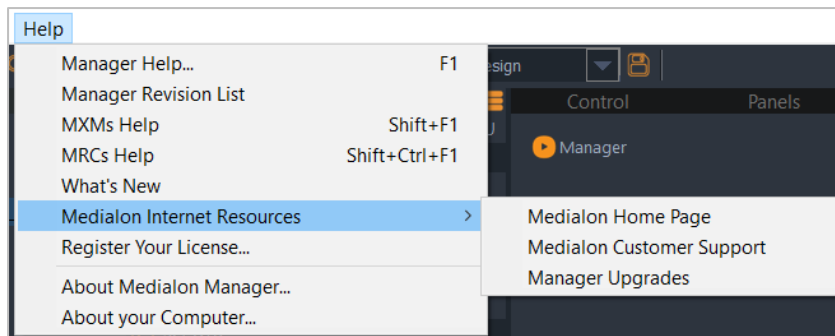
Closes all UserScreens that are currently open.

All Control Panels

Closes all Control Panels that are opened as separate windows; the Control Panels that are open in the Control tab stay open.

Help

This menu provides access to help files and information about your computer.



Manager Help (F1)

This is the reference guide of Medialon Manager V7 (this document's content).

Manager Revision List

This opens the installed revisions file:

C:/ProgramData/Medialon/Manager/Help/ReleaseInfoBase.html



MxMs Help (Shift+F1)

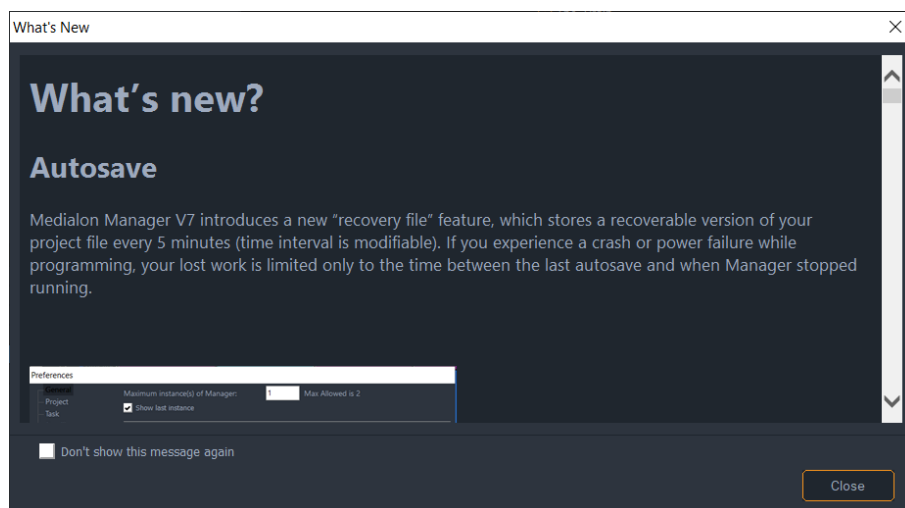
This window contains the list of installed MxMs and associated help documentation.

MRCs Help (Shift+Ctrl+F1)

This window contains the list of installed MRCs and associated help documentation.

What's New

Opens a panel containing Medialon new product features:



Medialon Internet Resources

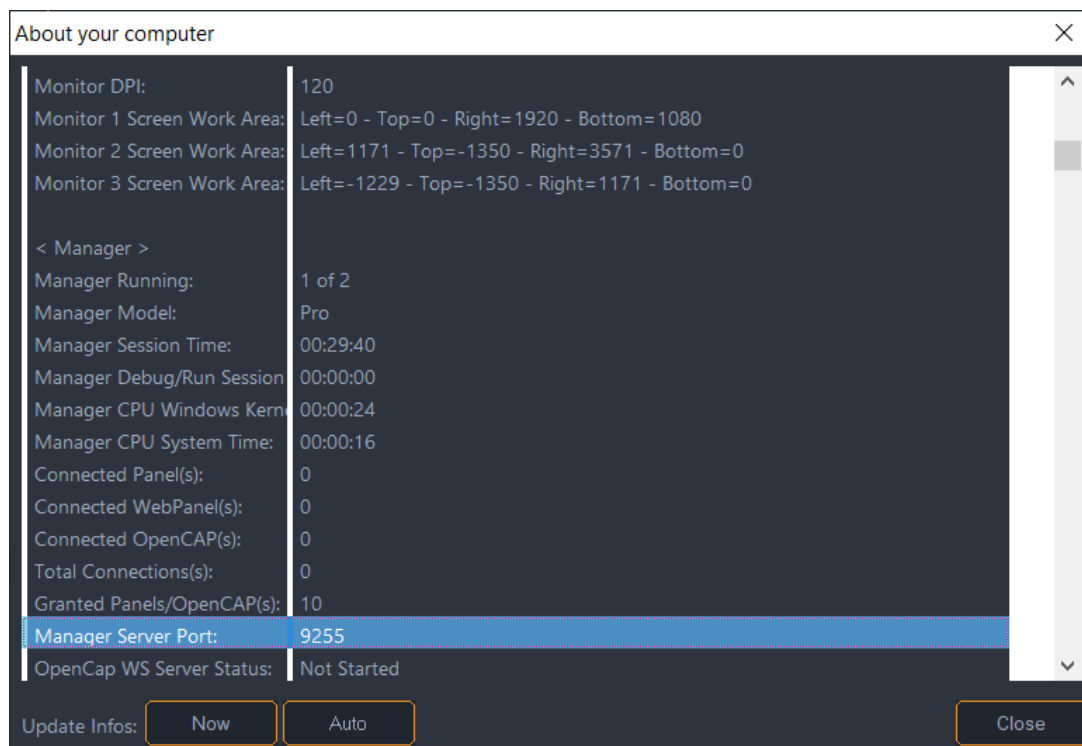
Gives you direct access to the 7thSense Medialon support portal (<https://portal.7thsense.one>).

About Medialon Manager

This is the splash screen of Medialon Manager V7 Software. Here you will find the number of the current version as well as the license number.

About your Computer

This window contains the main available information concerning your computer (system, memory, display graphic capabilities, etc.) and informs you of your work in Medialon Manager, as well as the number of Medialon Manager applications opened and the number of Medialon applications connected to the network.



It also contains information about the MxMs and MRCs installed on your system.

8.2 Drag and Drop Wizards

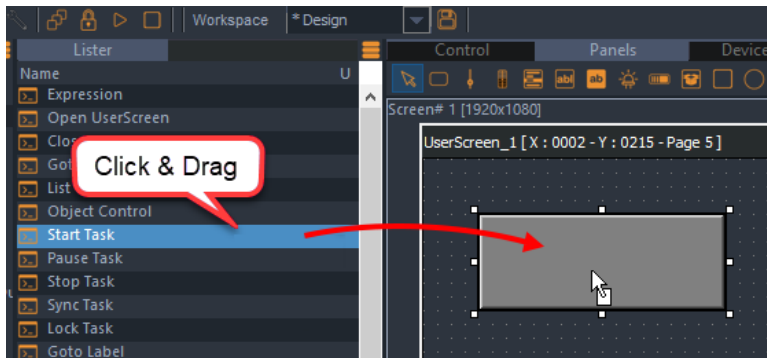
This feature of Medialon Manager V7 is the same as the drag and drop technique of adding cues to tasks. Drag and Drop wizards are useful and save a lot of time when programming your project.

8.2.1 Dragging onto Graphical Objects

With the Drag and Drop wizard in Medialon Manager V7, it is easy to create tasks linked to UserScreens objects.

Dragging Device Commands onto Graphical Objects

To create a task that sends a device command started by a graphical object, simply select the desired command in the device's Command Lister, and drag it onto the desired graphical object. The task is created with the correct start condition and the device command cue is inserted into the task.

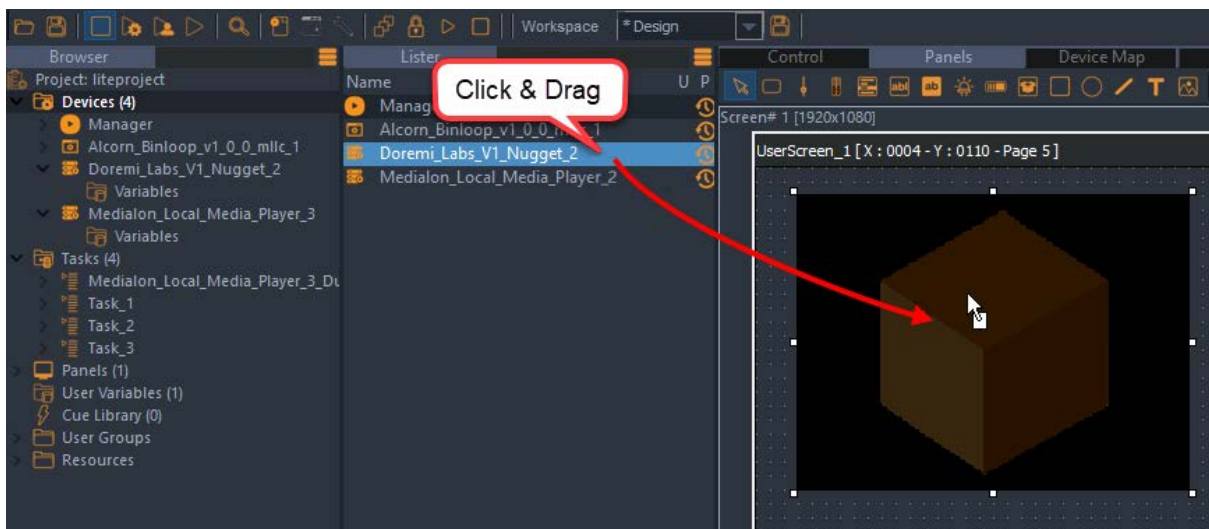


This function only creates step based tasks; should you need to create a time based task, you will have to create and link it 'manually'.

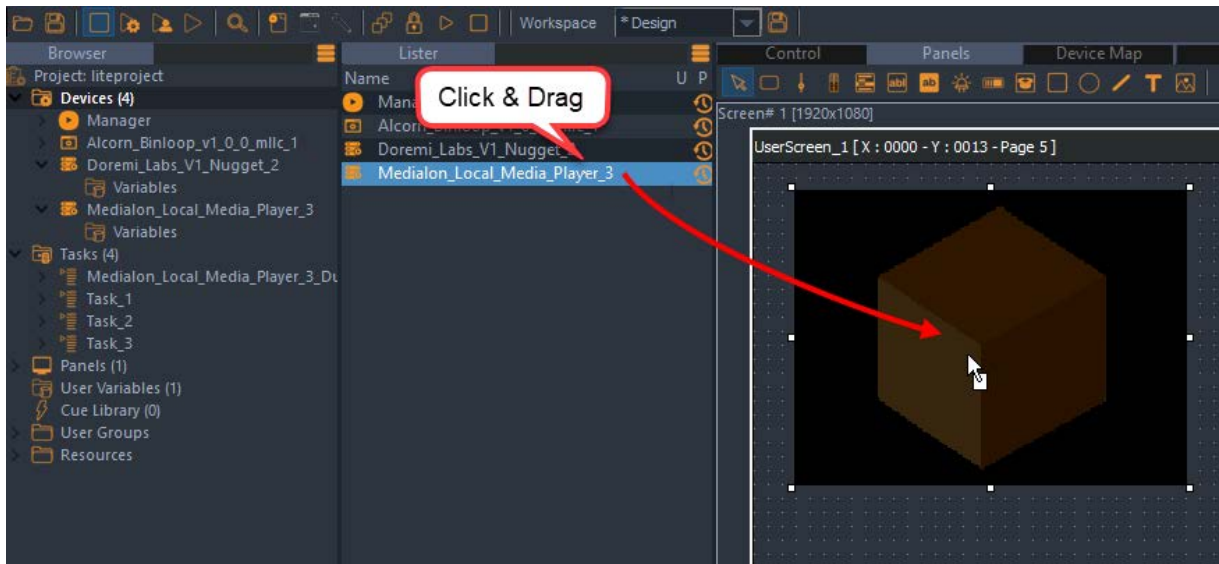
Dragging Devices on Container Objects

To open a Device Control Panel inside a Container, select a Device in the object browser or lister and drag and drop it onto a Container. This function is identical to creating a task that uses a Manager 'Device Control' cue.

Example: you wish to insert a Doremi Control Panel in your UserScreen.



The Device Control function is also used for the Device Display command; in this example, you will open a Local Media Player Display inside a container.

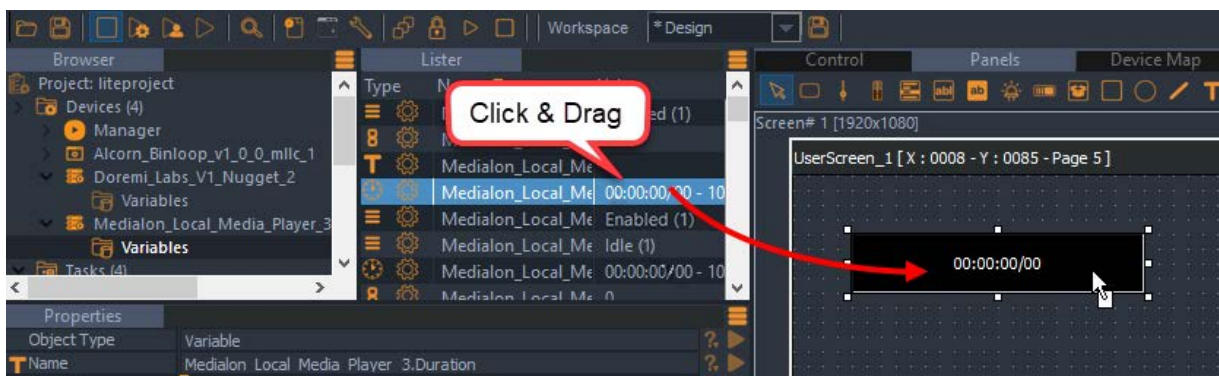


i Containers are on available in Local Panel (Manager V7).

Dragging Variables onto Graphical Objects

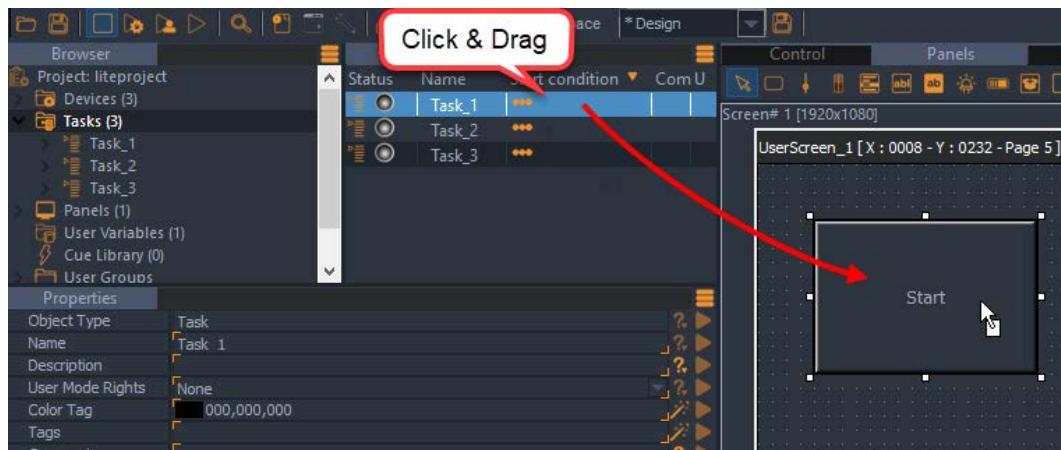
Similar to creating device commands, you may drag a variable onto an object to monitor the variable in the graphical object. This function is identical to the 'create a task to monitor a variable with this object' wizard, but is much faster to use.

For instance, you may want to create a display that shows the current time of the system. Create your display and edit its properties, then just drag the Manager.CurrentTime variable onto the display and you are good to go. As soon as you enter Debug or Run mode, the display shows you the current time.



Dragging Tasks onto Graphical Objects

To start a task with a graphical object, you can drag the task onto the object. This function is similar to the 'Add Start Condition to selected Task' wizard.



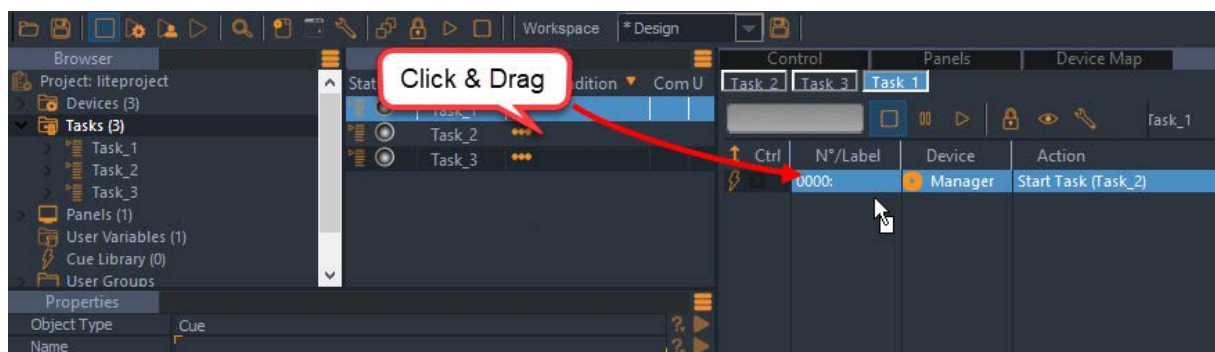
For instance, you wish to start your time based task with a button called 'Start'. Select 'Task_1' in the Lister and drag it onto 'Start' button. The start condition of Task_1 becomes:

`START.Status# & START.Status==1`

8.2.2 Drag and Drop on Tasks

Dragging a Task onto another Task

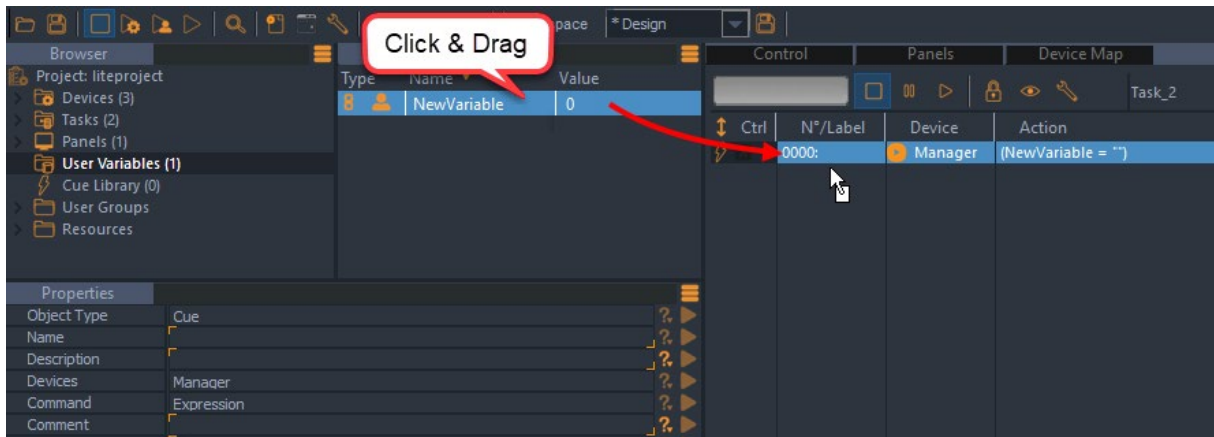
To start a task from another task, you simply drag one task into the other task. This action generates a Start Task cue and fills in the parameters.



For instance, select Task_1 in the Lister and drag it into Task_2 in the Task Editor; a new cue (Start Task(Task_1)) is inserted.

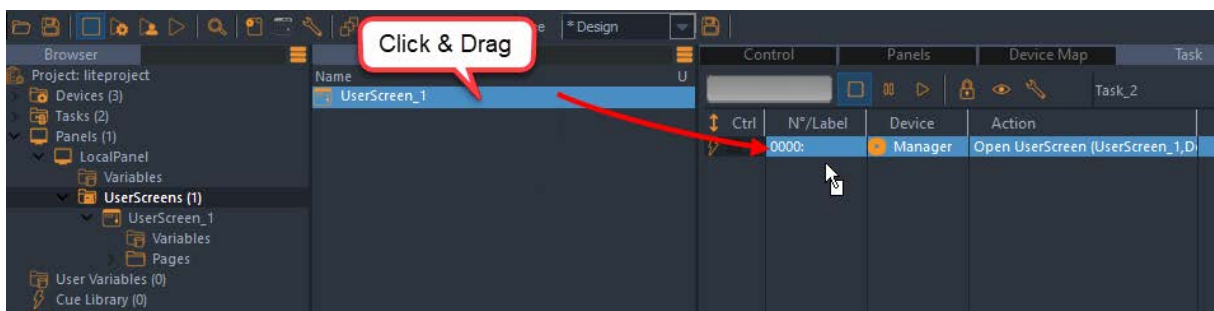
Dragging a Variable onto a Task

Select a variable in the Lister and drag it onto a task; a new expression cue is inserted (selected_variable = "").



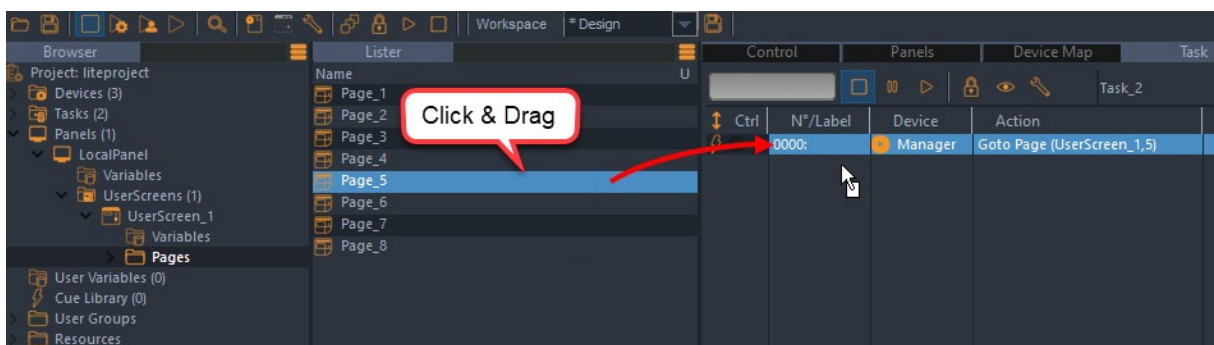
Dragging a UserScreen into a Task

Select a UserScreen in the Lister and drag it into a task. an 'Open UserScreen' command is inserted in the task with the selected UserScreen as parameter.



Dragging a Page into a Task

You can insert a 'Goto Page' when you drag a UserScreen Page from the lister onto a task.



Dragging a Cue into a Task

You can use drag and drop to move a cue into a task. This can be from a list of cues in the Lister tab or a cue from another task. Click, drag, and hover over the tab of the target task, and drop the cue into the appropriate place in the task.

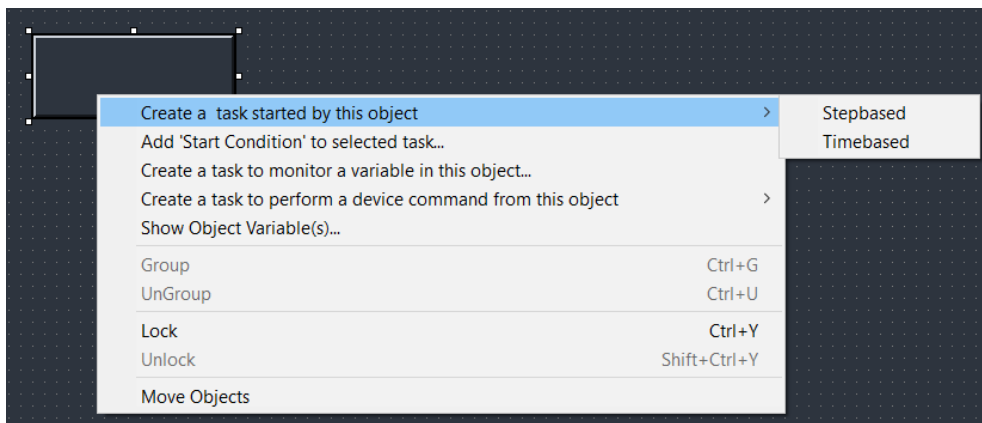
If you hold *Ctrl* key when you drop the cue, you copy it instead of moving it.

8.3 Task Creation Wizards

Dynamic Objects are capable of being linked to tasks in order to send commands, process information or display monitored variables available in your project. Medialon Manager V7 uses wizards that allow you to create tasks without any manual coding. There are several wizards at your disposal, each one designed for a specific situation.

8.3.1 Graphical Object Wizards

Right-clicking on a Dynamic Graphical Object opens a contextual menu that allows you to create a task related to that object.



Create a task started by this object

Creates either a step based or time based task related to the status change of the object.

The task that is created is automatically named with the name of the object and the action that it performs to trigger the task [objectname+action]. For instance, a task created from a button called 'PLAY' will take the name 'PLAYClick'.

The start condition of the created task is set to an evaluation expression that monitors the state of the object's trigger variable. Start conditions are case sensitive, and are different depending on the type of object that is to trigger the task. For instance, a button called 'PLAY' will create a task with the following start condition:

```
PLAY.Status# & PLAY.Status==1
```

A slider creates a different start condition:

```
SLIDER.Status#
```

If a task is already linked to this object, the contextual menu does not offer the possibility to create a task started by the object, but instead allows you to edit the task that is started by the object. This function provides very quick access to the tasks associated with objects so you can edit them without having to browse for them.

Add start condition to selected task

If a task is selected in the Lister, the right-click contextual menu allows you to update the start condition of the selected task by the selected object. The task is not modified or renamed, only the start condition is updated.

When a start condition already exists for the selected task, Medialon Manager asks if you would rather replace this start condition or to add the new condition to the existing start condition. In the second case, the new start condition is formatted in this way:

(Old Condition) | (New Condition)

Create a task to monitor a variable with this object

Dynamic graphical objects can monitor variables and update their status automatically. When you create a task to monitor a variable, the object reflects the variable's status. The created step based task has a start condition of:

[Selected Variable]#

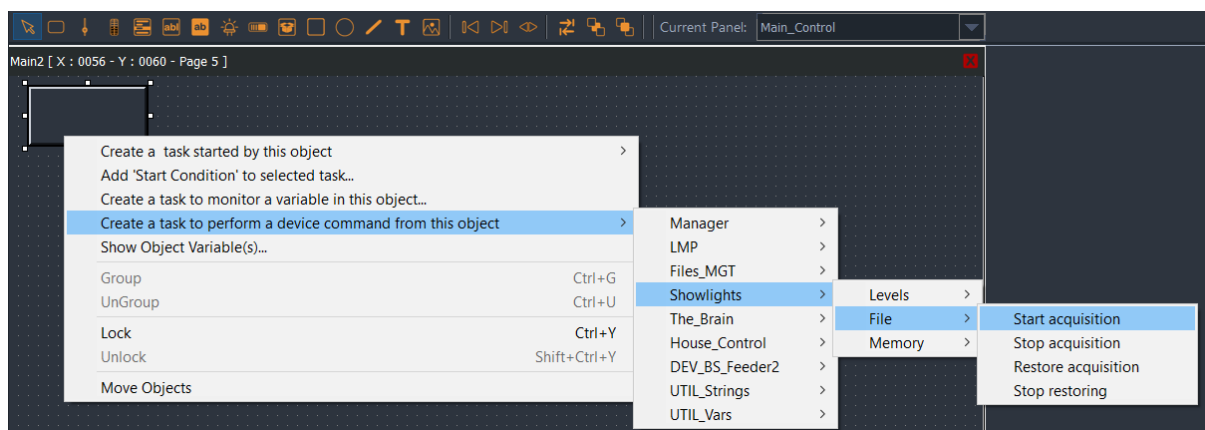
Furthermore, an assignment expression cue is inserted into the task that performs the following operation:

[Objectname.status]=[selected Variable]

The object must be able to take the value of the variable. For this reason, the Variable Selection Window only allows you to select a compatible variable to monitor. For instance, a button cannot monitor a String variable since the button's status variable is an integer.

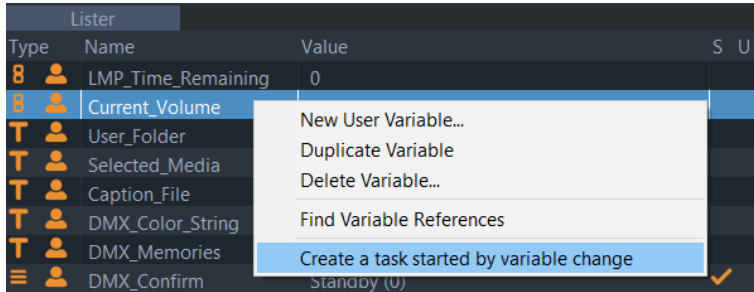
Create a task to perform a device command from this object

This option is very simple but useful and time saving. It allows you to create a step based task, with the correct start condition, that inserts a device command cue of your choice. Select the object and then right-click progressively for the options. All the devices available in the project are listed; simply browse through the list until you reach the desired one. If the command has parameters that need to be entered, you will need to do so in the Properties tab to send a valid command when the task is triggered.



8.3.2 Variable Wizards

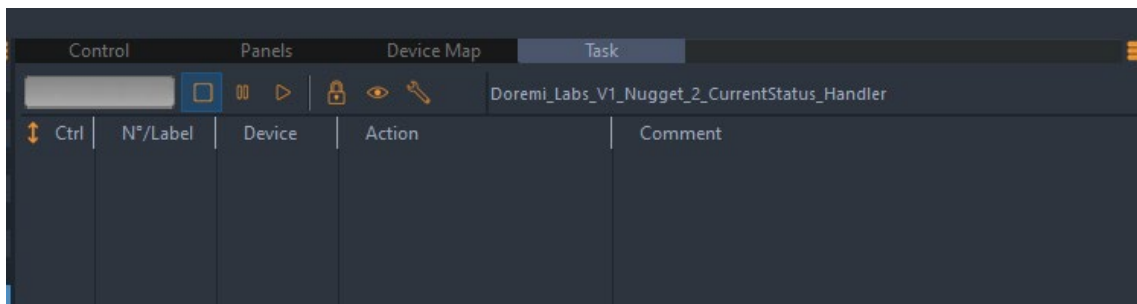
When a variable is selected in the Lister tab, right-clicking on the variable opens a contextual menu allowing you to create a task started by a change in that variable.



A task is created with the start condition's evaluation expression as follows:

[Selected Variable]#

The task is not linked to any graphical object nor will it have any cues automatically inserted. You must edit the task manually to suit your needs.



8.4 Keyboard Shortcuts

8.4.1 File main menu

Command	Shortcut	Action
New Project	Ctrl+N	Create a new empty project
Open Project	Ctrl+O	Open and load an existing project file
Save Project	Ctrl+S	Save the current project to disk
Debug	F2	Switch to Debug mode (from Stop mode)
User	Ctrl+F2	Switch to User Mode (from Stop mode)
Run	F3	Switch to Run mode (from Stop mode)
Stop	ESC	Switch to Stop mode
Print	Ctrl+P	Print the selected view
Quit	Ctrl+Q	Quit the application

8.4.2 Edit main menu

Command	Shortcut	Action
Undo	Ctrl+Z	Undo the last action
Redo	Shift+Ctrl+Z	Redo action previously undone
Cut	Ctrl+X	Cut the selected item or text into the clipboard
Copy	Ctrl+C	Copy the selected item or text into the clipboard
Paste	Ctrl+V	Paste the content of the clipboard
Select All	Ctrl+A	Select all items or text
Group	Ctrl+G	Group selected objects
Ungroup	Ctrl+U	Ungroup selected objects
Find...	Shift+Ctrl+F	Launch the Finder window
Lock	Ctrl+Y	Lock selected graphical object
Unlock	Shift+Ctrl+Y	Unlock selected graphical objects
Push to Back	Ctrl+B	Push selected graphical object to the background
Pull to Front	Ctrl+F	Pull selected graphical object to the foreground

8.4.3 Object main menu

Command	Shortcut	Action
Duplicate	Ctrl+D	Duplicate selected object
New Device	Ctrl+Ins	Create a new Device
New Variable	Shift+Ins	Create a new user variable
New Task	Ctrl+T	Create a new task
New Task Editor	Shift+Ctrl+T	Open an empty Task Editor

8.4.4 Windows main menu

Command	Shortcut	Action
Browser	F5	Show and focus on the Browser tab
Lister	F6	Show and focus on the Lister tab
Properties	F7	Show and focus on the Properties tab
Device Map	F8	Show and focus on the Device Map tab
Panels	F9	Show and give the focus to the Panels tab
Panels List	Ctrl+F5	Show the Panels node in the Browser and give the focus to the list of Panels in the Lister
Devices List	Ctrl+F6	Show the Devices node in the Browser and focus on the list of Devices in the Lister
Variable List	Ctrl+F7	Show the variables node in the Browser and focus on the list of variables in the Lister
Task List	Ctrl+F8	Show the Tasks node in the Browser and focus on the list of tasks in the Lister
Timecode Calculator	Ctrl+F11	Show the Timecode Calculator tool

8.4.5 Help main menu

Command	Shortcut	Action
Showmaster V2 Help	F1	Launch Medialon Manager V7 Reference Manual
MxM Help	Shift+F1	Launch MxMs' Help index

8.4.6 Workspace navigation

Command	Shortcut	Action
Area switch forward	F4	Switch between opened Areas in forward direction
Area switch backward	Shift+F4	Switch between opened Areas in backward direction
Tab switch forward	Ctrl+F4	Switch between opened Tabs inside an Area in forward direction
Tab switch backward	Shift+Ctrl+F4	Switch between opened Tabs inside an Area in backward direction

8.4.7 Browser Tab Navigation

Command	Shortcut	Action
Select Next	Down Arrow	Select the next item in the list
Select Previous	Up Arrow	Select the previous item in the list
Expand / Select Next	Right Arrow	Expand selected node or Select next node in the expanded node
Collapse / Select Previous	Left Arrow	Collapse selected node or Select previous node in the collapsed node
Enable Device Positrack	Ctrl + Shift + N	Enables Positrack for the selected Device
Disable Device Positrack	Ctrl + Shift + L	Disabled Positrack for the selected Device

8.4.8 Lister Tab Navigation

Command	Shortcut	Action
Select Next	Down Arrow	Select the next item in the list
Select Previous	Up Arrow	Select the previous item in the list
Create New Object	INS	Create a new Object of the same type as that listed in the Lister
Enable Device Positrack	Ctrl + Shift + N	Enables Positrack for the selected Device
Disable Device Positrack	Ctrl + Shift + L	Disabled Positrack for the selected Device

8.4.9 Properties Tab Navigation

Command	Shortcut	Action
Select Next	Tab	Select the next editable item in the list
Select Previous	Shift + Tab	Select the previous editable item in the list
Select Next	Mouse Wheel Down	Select the next editable item in the list
Select Previous	Mouse Wheel Up	Select the previous editable item in the list
Edit Selected	Right Arrow	Begin editing the selected parameter
Validate editing	Enter	When a parameter is edited, pressing Enter validates the new value
Cancel editing	Esc	When a parameter is edited, Esc cancels the edit
Test Command	Enter	When selected object is a command, execute the command
Select Next	Ctrl + Mouse Wheel Down	Select the next item in the list
Select Previous	Ctrl + Mouse Wheel Up	Select the previous item in the list

8.4.10 Task Editor Tab Navigation

Command	Shortcut	Action
Insert Cue	INS	Open the contextual menu to create a cue
Create Expression Cue	Ctrl+E	Create an expression cue
Create Wait Cue	Ctrl+W	Create a Wait cue (step based task only)
Create While/ EndWhile Cue	Ctrl+H	Create While/EndWhile cue (step based task only)
Create If/Else/EndIf Cue	Ctrl+I	Create If/Else/EndIf cue (step based task only)
Create WaitFor Cue	Ctrl+K	Create a WaitFor cue (step based task only)
Create OnExpression/Case Cue	Ctrl+J	Create OnExpression/Case/EndOnExpression cue (step based task only)
Create Case/EndCase Cue	Ctrl+J	Create new Case/EndCase cue inside an OnExpression/ EndOnExpression cue
Re-Insert last Cue	Alt+R	Re-Insert the last cue created
Create Message Cue	Shift+M	Create a Message cue
Create Comment Cue	Alt+Shift+C	Create a Comment cue
Create Countdown Cue	Alt+C	Create Countdown cue (time based task only)

Menus and Wizards

Command	Shortcut	Action
Add Cue to Library	Shift+A	Add selected cue to the Library
Insert Cue from Library	Shift+I	Open Library Menu to insert a cue
Deactivate Cue	Shift+S	Deactivate selected cue
Activate Cue	Shift+N	Activate selected cue
Toggle Pause Point	Alt+P	Toggle Pause Point
Edit Cue	ENTER	Edit the selected cue (default parameter)
Delete Cue	DEL	Delete the selected cue
Expand Cues	Shift+X	Expand in time selected cues (time based task only)
Compress Cues	Shift+C	Compress in time selected cues (time based task only)
Distribute Cues	Shift+D	Distribute regularly in time selected cues (time based task only)
Offset Cue Time	+	or Enter an offset to apply to the time position of selected cue (time based task only)
Offset/Modify Cue Time	=	Add/substitute an offset or modify the time position of selected cue (time based task only)
Select next Cue	Right Arrow	Select next cue in the timeline view
Select previous Cue	Left Arrow	Select previous cue in the timeline view
Select next Cue	Down Arrow	Select next cue in the List view
Select previous Cue	Up Arrow	Select previous cue in the List view
Select next Track	Down Arrow	Select the next track in the timeline view
Select previous Track	Up Arrow	Select the previous Track in the timeline view
Navigate from Tab to Tab	Ctrl+Tab	Navigation from Tab to Tab inside the Task Editor
Move Cue forward by 1/10 s	Alt+Right Arrow	Move selected cue forward in the timeline by 1/10 second
Move Cue backward by 1/10 s	Alt+Left Arrow	Move selected cue backward in the timeline by 1/10 second
Move Cue forward by 1/100 s	Ctrl+Alt+Right Arrow	Move selected cue forward in the timeline by 1/100 second
Move Cue backward by 1/100 s	Ctrl+Alt+Left Arrow	Move selected cue backward in the timeline by 1/100 second
Move Cue to next Track	Alt+Down Arrow	Move cue to the next Track in the timeline
Move Cue to previous Track	Alt+Up Arrow	Move cue to the previous Track in the timeline
Move Time Cursor forward 1/10 s	Shift+Right Arrow	Move the Time Cursor forward 1/10 second in the timeline
Move Time Cursor backward 1/10 s	Shift+Left Arrow	Move the Time Cursor backward 1/10 second in the timeline
Select All Cues	Ctrl+A	Select all cues in the List view
Select All Cues in the current Track	Ctrl+A	Select all cues in the selected rack in timeline view
Select All Cues in Timeline	Ctrl+Shift+A	Select all cues in the timeline
Select Cues from Start	Shift+F	Select all cues from the start of the timeline to cursor position in the active track
Select Cues to End	Shift+T	Select all cues from cursor position to the end of the timeline in the active track
Select Cues from Start	Ctrl+Alt+F	Select all cues from the start of the time based task to cursor position in all tracks
Select Cues to End	Ctrl+Alt+T	Select all cues from cursor position to the end of the time based task in all track elect First Cue HOME Select the first cue of the task (time based task only)

Command	Shortcut	Action
Select Last Cue	END	Select the last cue of the task (time based task only)
Deselect all	ESC	Deselect all cues (time based tasks only)
Start/Pause	SPACEBAR	Start or Pause the task
Make Task Editor Text Bigger	Ctrl+ + (NumPad Plus)	In List View of timeline and in step based tasks
Make Task Editor Text Smaller	Ctrl+ -(NumPad Minus)	In List View of timeline and in step based tasks

8.4.11 Time X window navigation

Command	Shortcut	Action
Start/Insert Cue	SPACEBAR	Start Time X distribution or Insert the next cue
End of Time X	ESC	Terminate Time X distribution

8.4.12 UserScreens manipulation

Command	Shortcut	Action
Move Object Left	Alt+Ctrl+Left	Arrow Move left the selected Object
Move Object Right	Alt+Ctrl+Right	Arrow Move right the selected Object
Move Object Up	Alt+Ctrl+Up	Arrow Move up the selected Object
Move Object Down	Alt+Ctrl+Down	Arrow Move Down the selected Object
Resize Object (decrease width)	Alt+Shift+Left	Arrow Decrease width of the selected Object
Resize Object (increase width)	Alt+Shift+Right	Arrow Increase width of the selected Object
Resize Object (decrease height)	Alt+Shift+Up	Arrow Decrease height of the selected Object
Resize Object (increase height)	Alt+Shift+Down	Arrow Increase height of the selected Object
Color Picker	Ctrl+Alt+ Mouse Click	Add pointed object color to custom color

8.5 Action on Double-click

The use of the double-click is widely implemented in Showmaster Editor and can greatly accelerate your programming speed.

8.5.1 Cue double-click

Opens the default parameter in the properties tab for editing.

8.5.2 Variable double-click

In the Lister, opens the properties for editing (if possible).

8.5.3 Device Map double-click

Double-clicking the background will open the bitmap property.

8.5.4 Task double-click

- *Name or Browser Icon*: Opens the task in an editor.
- *Start Condition*: opens the Edit Expression window of the start condition.
- *Comments*: Opens the comment property.

8.5.5 Ctrl + Task double-click

In the Browser or on the name in the Lister, opens the Task Setup window.

8.5.6 Device double-click

Opens the device commands in the Lister.

8.5.7 Ctrl + Device double-click

Opens the device setup window.

8.5.8 Command double-click

Opens the default parameter in the properties tab.

8.5.9 Panel double-click

Opens the Panel in the Panels tab.

8.5.10 UserScreen double-click

Opens the UserScreen in the Screen tab.

8.5.11 User Group double-click

Opens the content of the group.

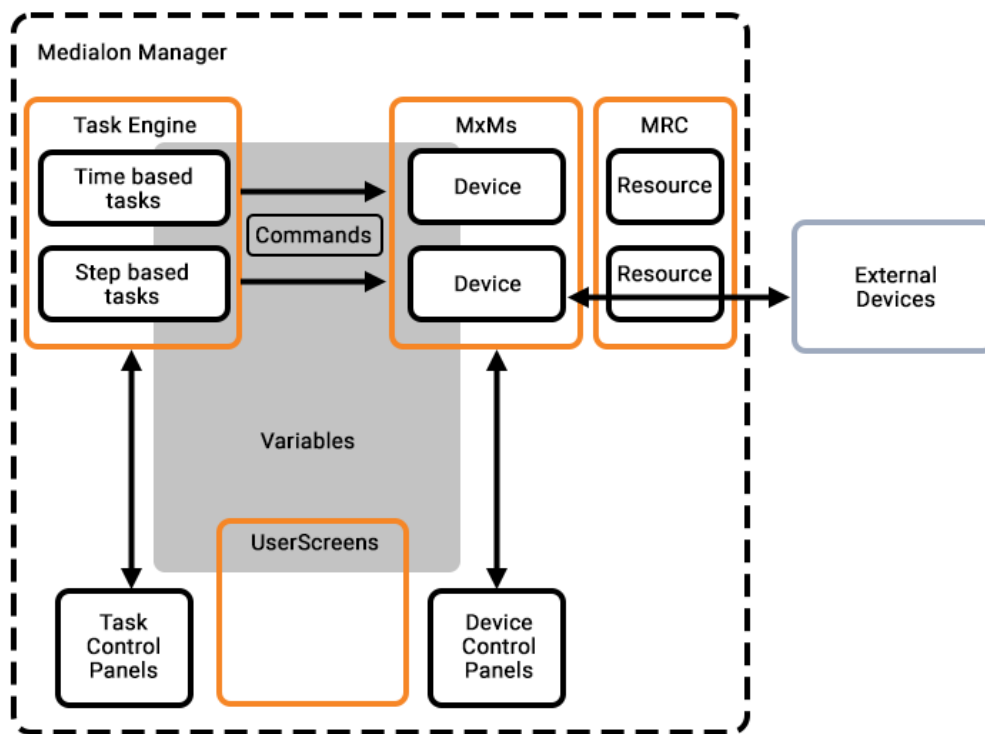
9 Manager V7 Specific Features

This section describes functions specific to Medialon Manager V7 software.

9.1 Overview of Medialon Manager

Medialon Manager V7 is a Windows® based control software. Medialon Manager can control any audiovisual equipment using any protocol and network. Being software based it integrates natively with IT, opening a wide range of applications. Control of audio and video streaming, links to database, media management, and facility wide control are easy to accomplish with Medialon Manager V7.


9.1.1 Medialon Manager V7



Medialon Manager V7 is the most feature-rich version of Medialon Manager. It gives access to all programming features, like graphical user interfaces (UserScreens), external networked resource connections, database connections, links to PLCs and Building Management Systems, and more.


9.1.2 Medialon Manager Modes

Stop Mode

In this mode , Manager V7 does not communicate with any external equipment nor does it perform any internal tasks. Stop Mode is the default programming mode where your project is built. Devices, tasks, UserScreens and variables can be created, modified, and deleted freely.


Pressing the *Esc* key or clicking on the Stop Project button returns you to this mode.

Debug Mode

In Debug Mode , Medialon Manager fully communicates and interacts with external devices. Programming windows such as the Object Browser, Object Properties and Task Editors are still accessible. This allows you to see what is happening internally while your project runs, allowing you to make adjustments without stopping the project.


Debug Mode is accessible from Stop Mode by typing the *F2* key or clicking on the Debug Project button.

User Mode

Medialon Manager's User Mode  gives the opportunity for the programmer to decide which objects are accessible to the user (tasks, devices, command for devices, variables...) but still gives the User flexibility to adjust some parts of the programming while masking other parts of the programming from the User. User Mode works like the Debug Mode, only with selected objects.

User Mode is accessible from Stop Mode by typing *Ctrl+F2* or clicking on the User Mode Project button.

Run Mode

Run Mode  is the final runtime mode of a project. Medialon Manager fully communicates and interacts with external devices and hides all the programming windows. Only the UserScreens or the Device Control Panels and Task Control Panels are displayed.

Run Mode is accessible from Stop Mode by typing the *F3* key or clicking on the Run Project button.

9.1.3 Medialon Manager V7 License Options

Medialon Manager V7 is a program protected by a hardware license and will only run with the appropriate license dongle connected to the USB port of your PC.

Medialon Manager V7 Pro

Medialon Manager V7 Pro is designed for any show and media control application, from simple stand alone to large networked systems.

Medialon Manager V7 Pro offers multiple time based tasks, step based tasks, and customizable control panels (UserScreens).

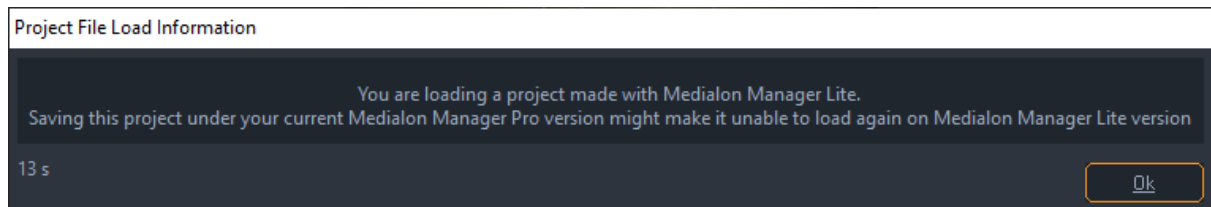
Medialon Manager V7 Lite

Medialon Manager V7 Lite is designed for small- to medium-size permanent installations.

Though limited in capacity, it offers all Medialon Manager's control and programming power at an affordable price.



A project saved with a Medialon Manager Lite can be opened with a Medialon Manager Pro. Any modifications could change the project to a Medialon Manager Pro project file.



To prevent this, you can launch a Medialon Manager Pro as Medialon Manager Lite by using key combination *CTRL + ALT* while the software starts up.

Medialon Manager V7 Panel

Medialon Manager V7 Panel is specifically designed to implement remote control panels, typically for deployment on touch screen computers. This free version of Medialon Manager cannot control external devices but is able to connect and share data with other Medialon Managers on the network.

9.2 Resources

Once the required MxMs are set up in Manager V7, you can create the devices you want within your program. Devices are the machines, programs, or services that Medialon Manager will control, via the MxMs and the hardware interfaces connected to it. An MxM can host several devices. For example, you may create as many Doremi devices as you wish using the one Doremi MxM.

9.2.1 Resource Management (MRC)

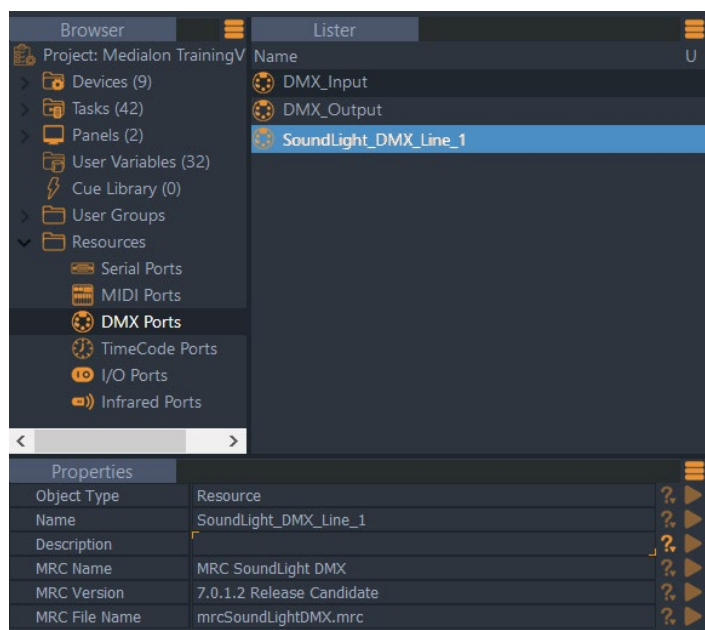
Medialon Manager has always been hardware independent; it can use virtually any hardware to control devices. Independently from the MxMs, Medialon Manager V7 uses resources to physically connect to devices.

Medialon Resource Connector (MRC) manages and configures the necessary resources for your projects. Resources are the means by which device drivers (MxMs) communicate with physical devices. Resources can be of the following types:

- Serial Ports
- MIDI Ports
- DMX ports
- TimeCode Ports
- Input/Output Ports
- Infrared Ports

All resources installed in your PC are automatically scanned when Medialon Manager starts, and they are included when your project loads.

Medialon Manager V7 can also manage external resources, such as serial I/O devices or networked resources (Art-Net, Advantech ADAM I/Os, AMX NetLinx Serial and I/O, Global Caché, etc.)

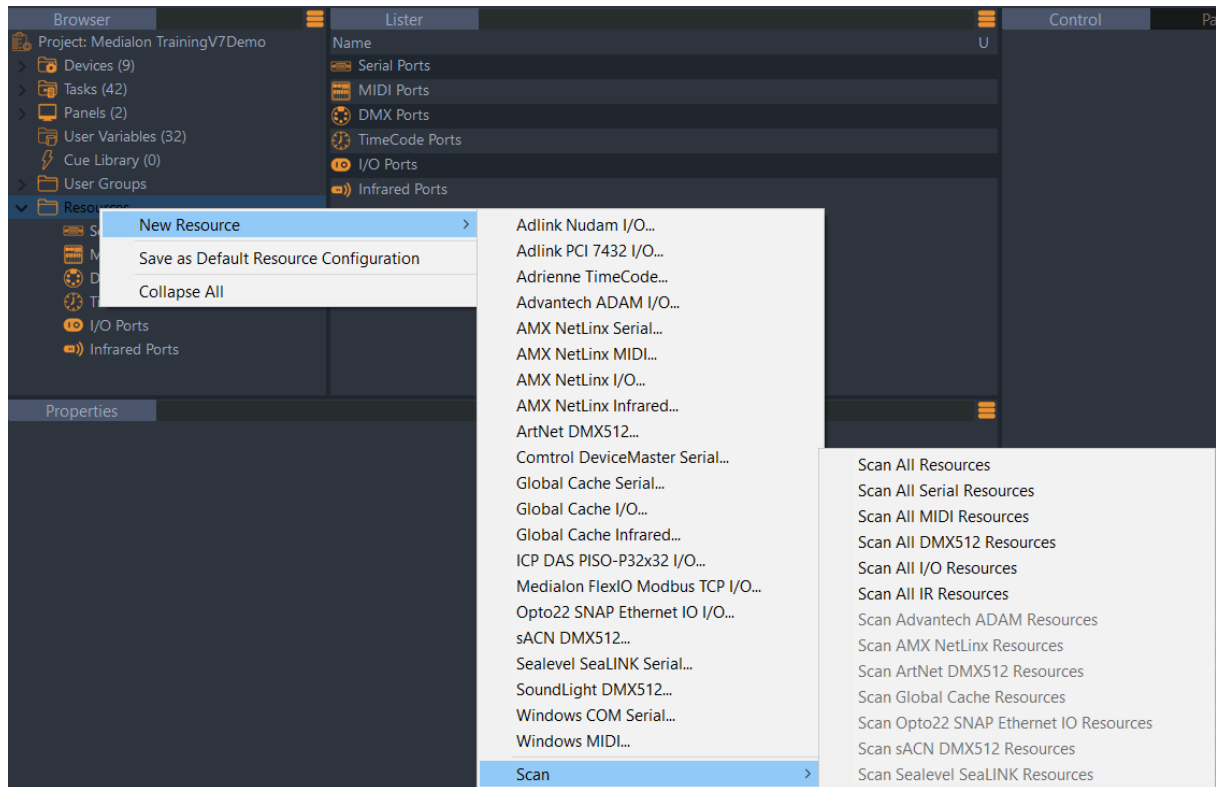


The MRCs can also define 'virtual' resources that are not yet available but will eventually be in the final installation.

Add New Resource

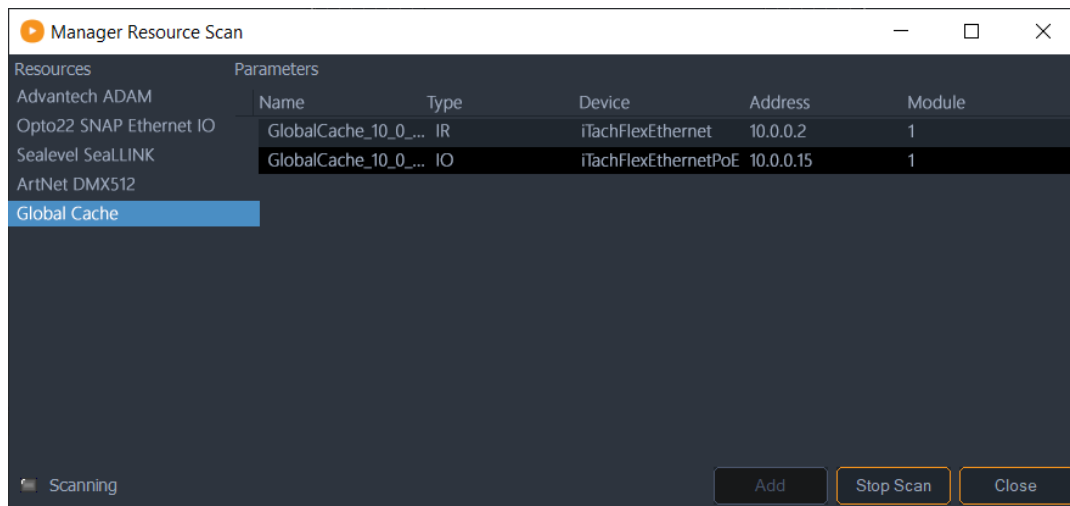
The Resources folder in the Browser tab displays the different types of resources available for your project. To add a new resource, right-click on the Resources node and select the New Resource you want to add or scan for resources.

Manager V7 Specific Features



All resources scan

The All resources scan will detect all available resources present on the network.

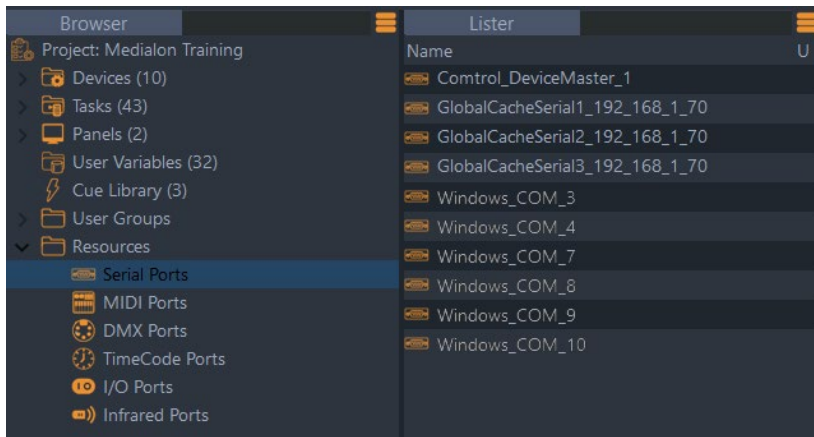


Select a resource and click on *Add* button to add this resource to your project.

Serial Ports

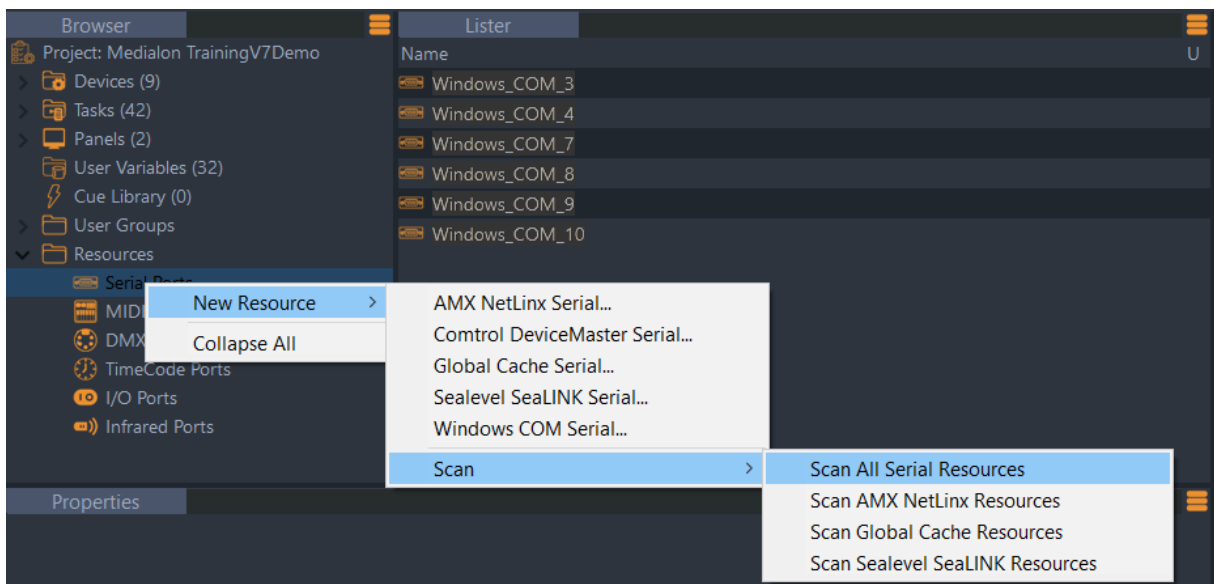
Click on the Serial Ports node to display the resources that were present in your project when Manager V7 started.

Manager V7 Specific Features



Different types of Serial resources can be mounted in your project.

You may add Serial ports to your project by right-clicking the 'New Resource' menu and choosing which port you wish to add.



Serial Ports are used by (non-exhaustive list):

- MxM Medialon Low Level Communicator
- MxM Doremi Labs V1 and Nugget
- MxM Sony RS422
- MxM Barco DLP VP
- MxM Analog Way iX-Series
- etc.

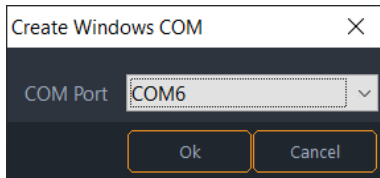


Refer to [\[Annexes / Medialon Resource Connector\]](#) for non-Windows Resources.

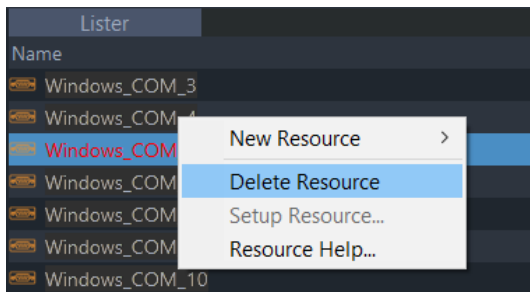
Windows COM Resources

All Windows COM Ports installed in your PC are scanned and included in your project's resources when Medialon Manager starts.

You may add 'virtual' COM ports to your project by right-clicking the 'New Resource/Windows COM Serial' menu and choosing which port you wish to add (up to 255).

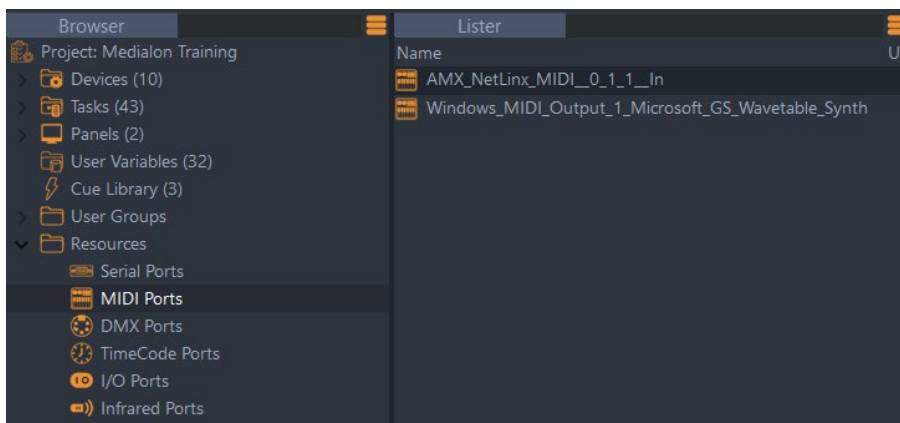


If you wish to delete a 'virtual' COM port you may do so, but the 'Real' ports cannot be removed from a project.



MIDI Ports

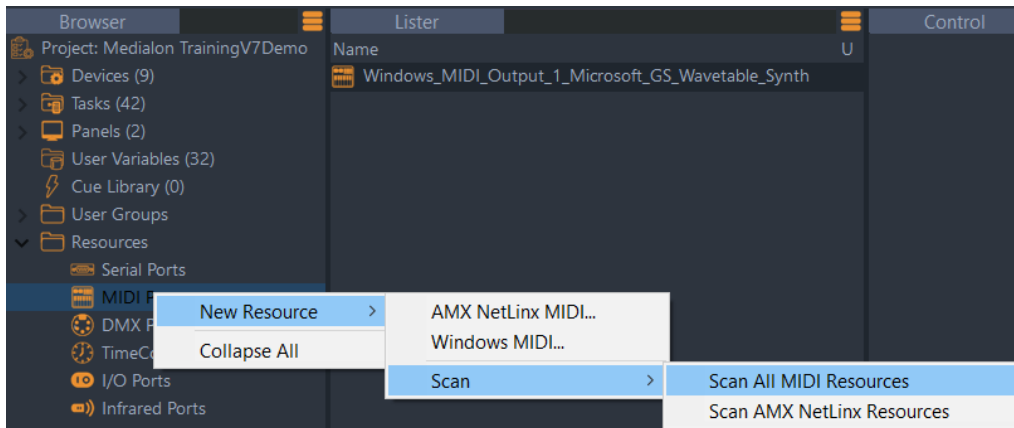
Click on the MIDI Ports node to display the resources that were present in your project when Manager V7 started.



Different types of MIDI resources can be mounted in your project.

You can add a new MIDI Port by right-clicking the 'New Resource' menu, then right-clicking progressively for further options.

MIDI Ports are used by (non-exhaustive list):



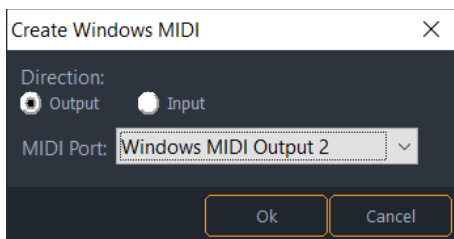
- MxM Medialon Basic Midi IN/OUT
- MxM Medialon MIDI Machine Control
- MxM Medialon MIDI Show Control
- MxM Medialon Low Level Communicator
- etc.

i Refer to [\[Annexes / Medialon Resource Connector\]](#) for non-Windows Resources.

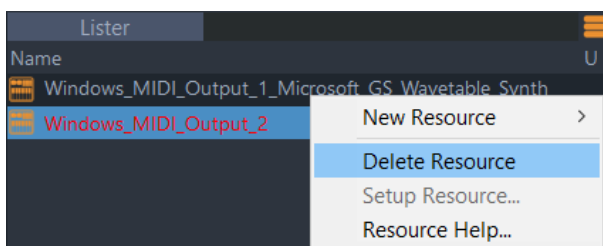
Windows MIDI Resource

MIDI Input and Output Ports installed in your PC are scanned on startup are automatically shown as available.

MIDI Inputs and Outputs can be added. You may create up to 64 Windows MIDI Input or Output Ports (this is a limitation of Windows). Additional MIDI resources are available beyond this limit if you use AMX Netlinx MIDI.



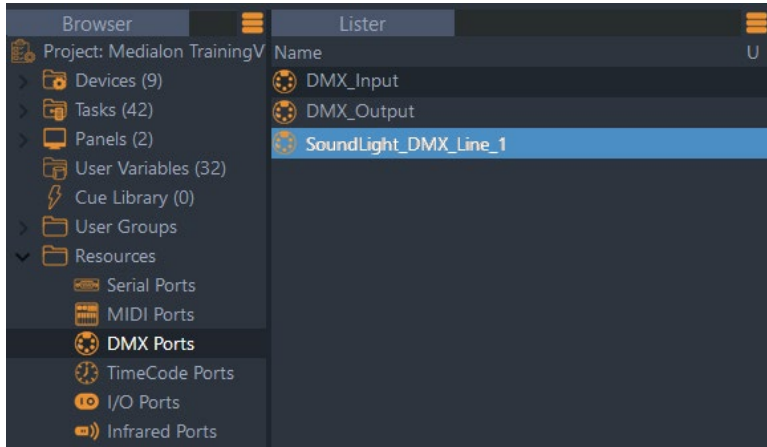
If you wish to delete a 'virtual' MIDI port, you may do so, but the 'Real' ports cannot be removed from a project.



DMX Ports

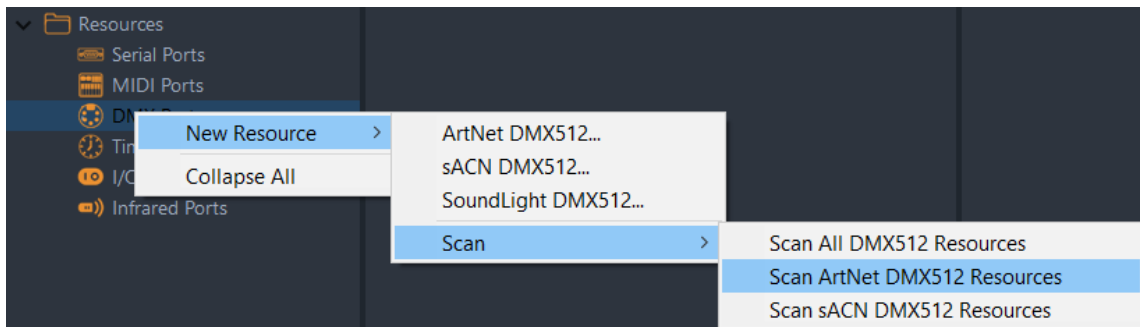
DMX Resources are managed by DMX MRCs. If hardware DMX resources are scanned on startup, they appear as available.

Click on DMX Ports node to display DMX resources.

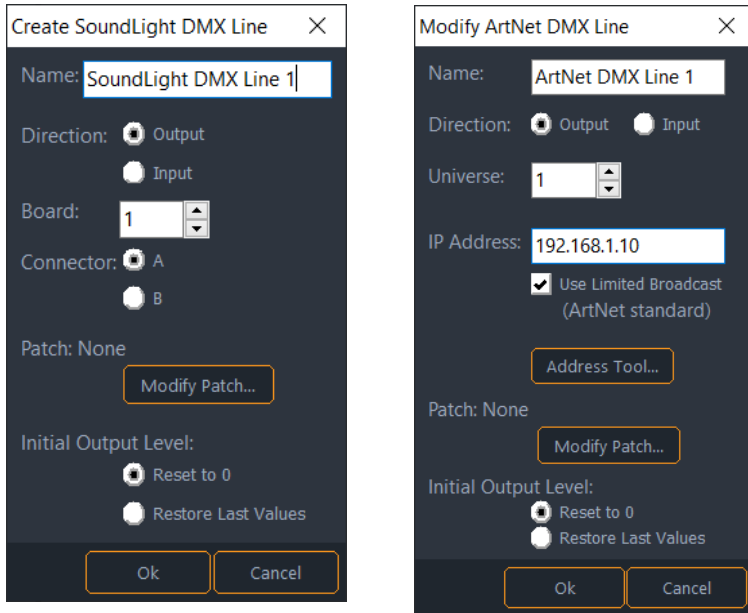


Different types of DMX resources can be mounted in your project.

If you create a 'Virtual' resource, the setup window automatically opens for you to configure it.

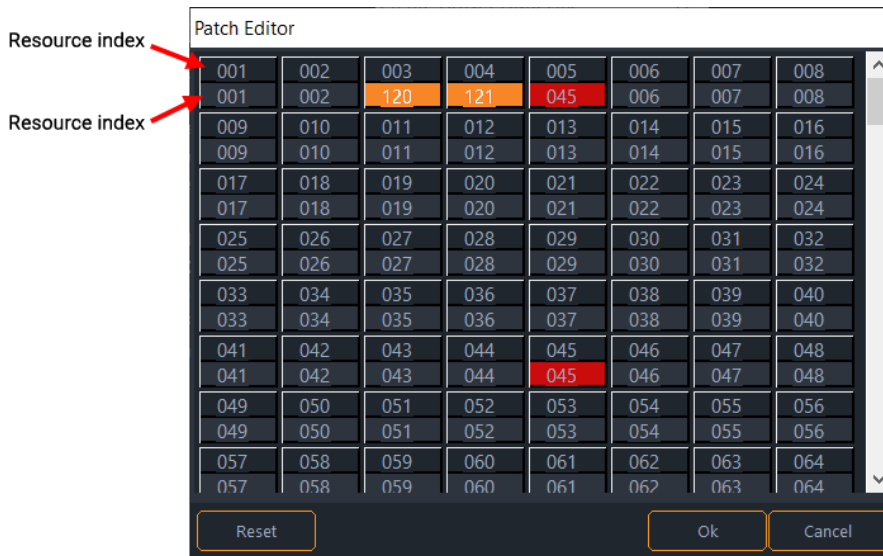


If you wish to configure the DMX Resources, select the resource in the Lister and use the right-click menu.



DMX Resources have common configuration.

- *Name*: Name given to the resource.
- *Direction*: Determines whether the resource is used for Input or Output.
- *Patch*: DMX channels can be redirected through a Patch modified to your needs.



Orange channels are patched. A red channel indicates a conflict with another index. Two or more indexes are patched to the same channel.

- *Initial Output Level*: You may choose to Reset all the Outputs to 0 when the project starts, or to keep the last values that were sent.

DMX Resources can be used by MxM Medialon DMX512 and MxM Medialon Script.



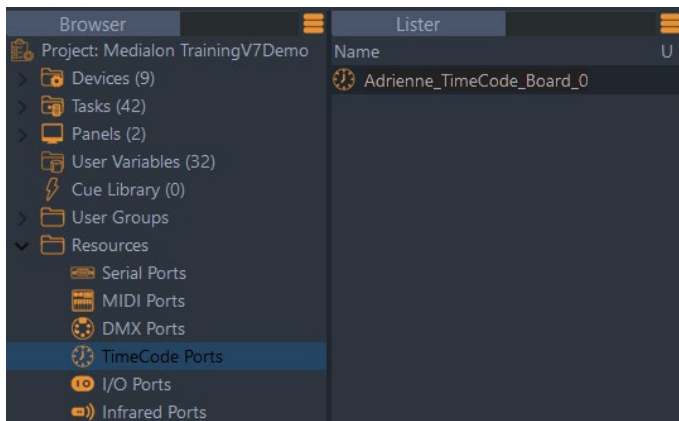
Refer to [\[Annexes / Medialon Resource Connector\]](#) for DMX Resources description.

TimeCode Ports

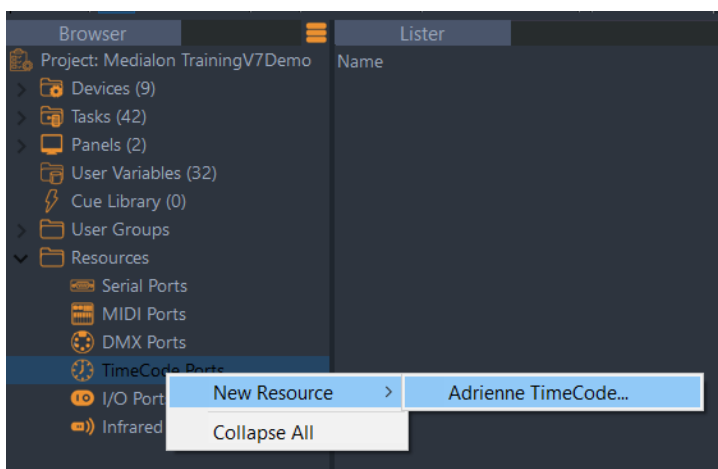
Medialon Manager can use Timecode Resources in your project.

MxM Medialon Timecode allows Manager V7 to read or generate Timecode.

Click on the TimeCode Ports node in Resources to display the Timecode resources present in your project when Manager V7 started.



You may add Timecode ports to your project by right-clicking the 'New Resource' menu and choosing which port you wish to add.



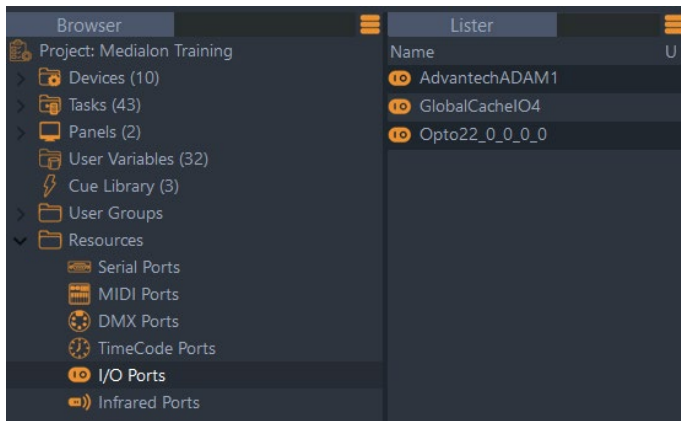
 Refer to [\[Annexes / Medialon Resource Connector\]](#) for TimeCode Resources description.

Input/Output Ports

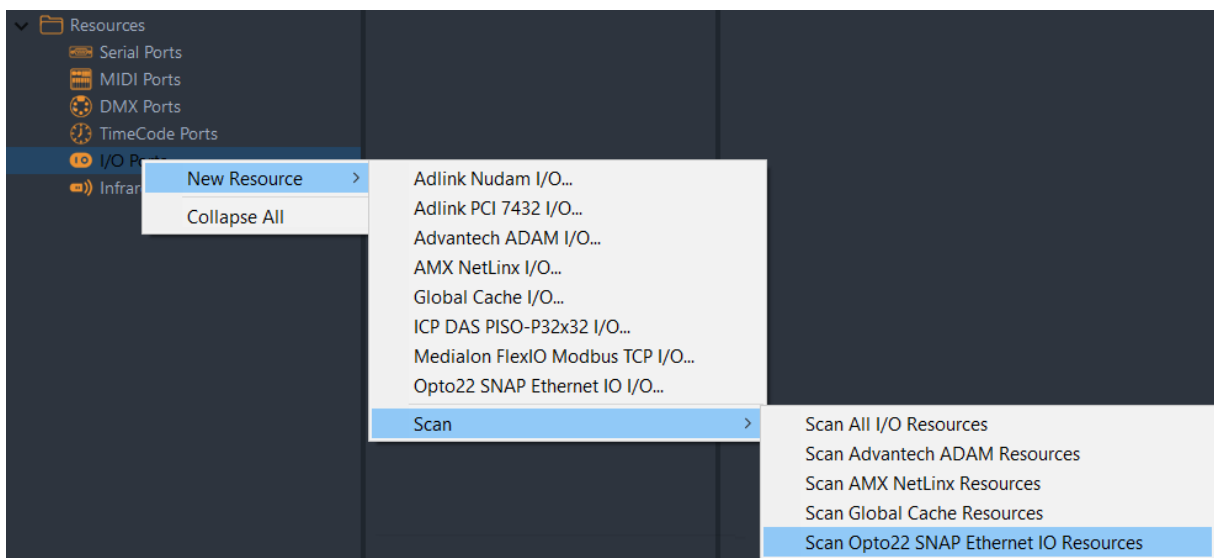
Medialon Manager V7 can manage Digital and Analog Input/Output resources such as relays and contact closures, analog outputs, etc.

Click on the I/O Ports node to display the I/O hardware resources present in your project when Manager V7 started.

Manager V7 Specific Features



You may add I/O ports to your project by right-clicking the 'New Resource' menu and choosing which port you wish to add.

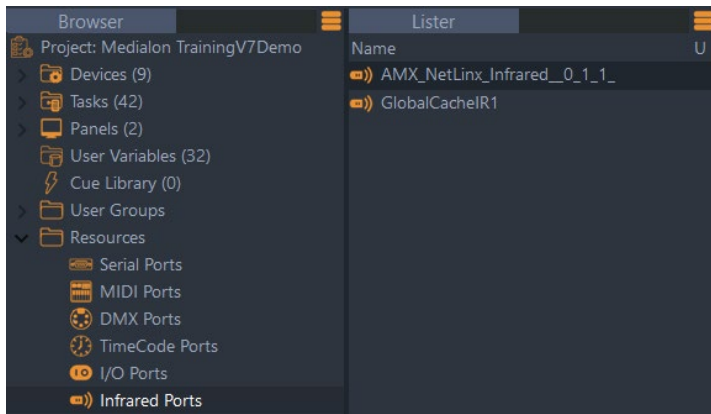


I/O Ports are used by MxM Medialon IO.

i Refer to [\[Annexes / Medialon Resource Connector\]](#) for I/O Resources description.

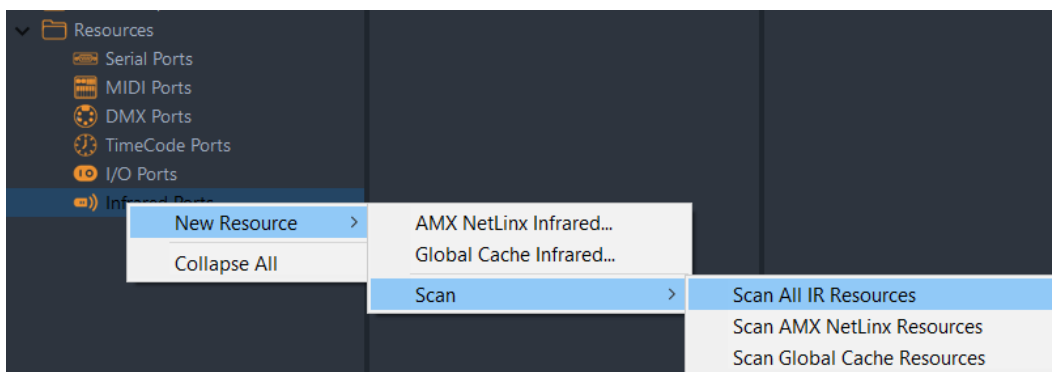
Infrared Ports

Infrared resources allow Medialon Manager V7 to generate Infrared signals. Click the Infrared Ports node to display the resources present in your project.



You may add Infrared ports to your project by right-clicking the 'New Resource' menu and choosing which port you wish to add.

Infrared Ports are used by MxM Medialon Infrared.



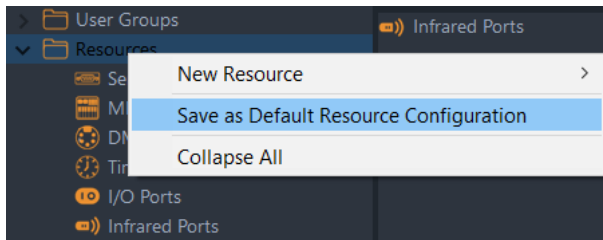
i Refer to [\[Annexes / Medialon Resource Connector\]](#) for a description of Infrared Resources.

Saving Resource Configurations

Once you have set up the resources, they are saved in your project, not in the Medialon Manager application itself. The 'Virtual' resources can be different for each project, and automatically become 'Real' if the necessary hardware is available on startup. If you open a project and do not have the available resources, the Resources become 'Virtual' so the project can be worked on 'offline' using a different system.

You may also choose, in MxM Setup, to use the Unspecified option for the resource assignment. This allows you to program your project before configuring the final resources that will be available.

If you will need to use the same resource configuration for different projects, you may find it useful to save a Default Resource Configuration with the desired resources being simulated.



This Default Resource Configuration will be recalled each time you start Medialon Manager V7.



'Virtual' resources are not connected to any physical hardware, but they act as real resources for your Medialon Manager V7 project.

9.2.2 MRC Help

Medialon Manager includes a Help file for each MRC.

Select the resource in the Lister tab: use the right-click menu to open the MRC Help. Each MRC is provided with a complete Help file.

The Help document contains:

- Overview of the real Resource
- Resource Creation and Setup
- Release Notes of the MRC

9.3 Medialon Watchdog

Medialon Watchdog is an external program which regularly monitors and controls the proper running of Medialon Manager and can restart Medialon Manager software or reboot the PC if Manager freezes up, crashes, or becomes unresponsive. Medialon Watchdog is able to send a failure report by email.

9.3.1 Watchdog Overview

The operating principle of 'Watchdog' is as follows:

- Medialon Manager 'announces' its presence to 'Medialon Watchdog' and how often it will report its activity to it.
- According to the frequency of these reports, Medialon Manager sends a signal to 'Medialon Watchdog' indicating its proper operation.
- If a problem occurs in Medialon Manager, 'Medialon Watchdog', which will have ceased to receive these signals, proceeds with an application restart or PC reboot according to the chosen options.

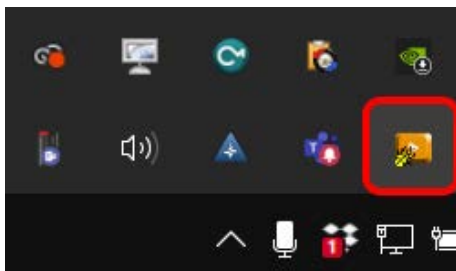


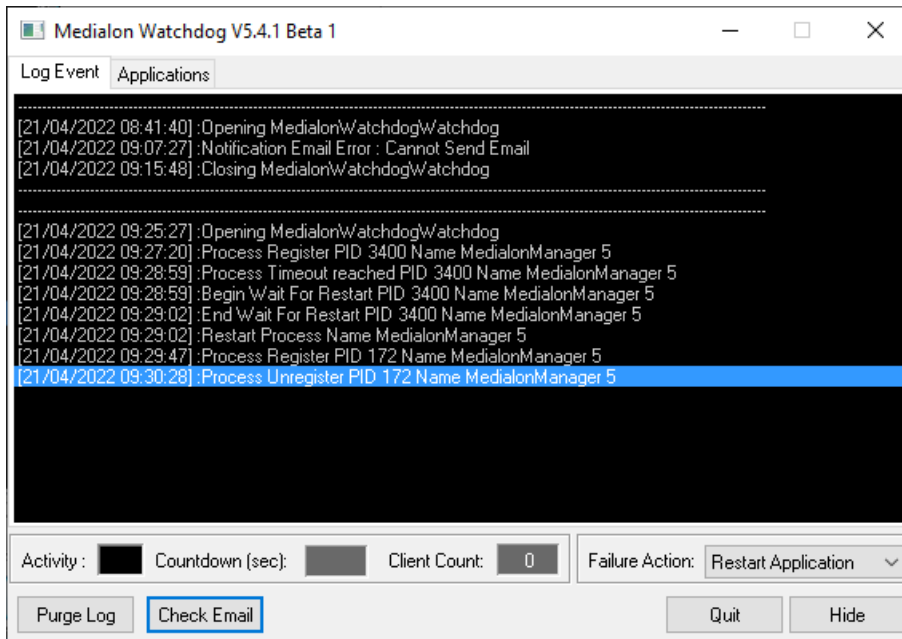
Medialon Watchdog is automatically installed during Medialon Manager installation. It is Active or Inactive according to File > Preferences > Watchdog.

9.3.2 Medialon Watchdog Window

Note: Medialon Watchdog is started at Windows startup. It is not closed with Medialon Manager.

To open the Medialon Watchdog window, double-click the Medialon Watchdog icon in the system tray.





A Quit button allows you to stop Medialon Watchdog. The Hide button only closes the Medialon Watchdog window.

Failure Action

Defines the behavior of Medialon Watchdog when one of the monitored applications freezes.

Restart Application

Medialon Watchdog will restart the Medialon Manager application when its activity is no longer detected.

Restart Computer

Medialon Watchdog will start the computer's reboot procedure when Medialon Manager is no longer sending the watchdog activity updates. This option stops all running applications and reboots the PC.

Activity

LED blinks when an application is monitored. Frequency depends on Medialon Manager's timeout settings.

Countdown

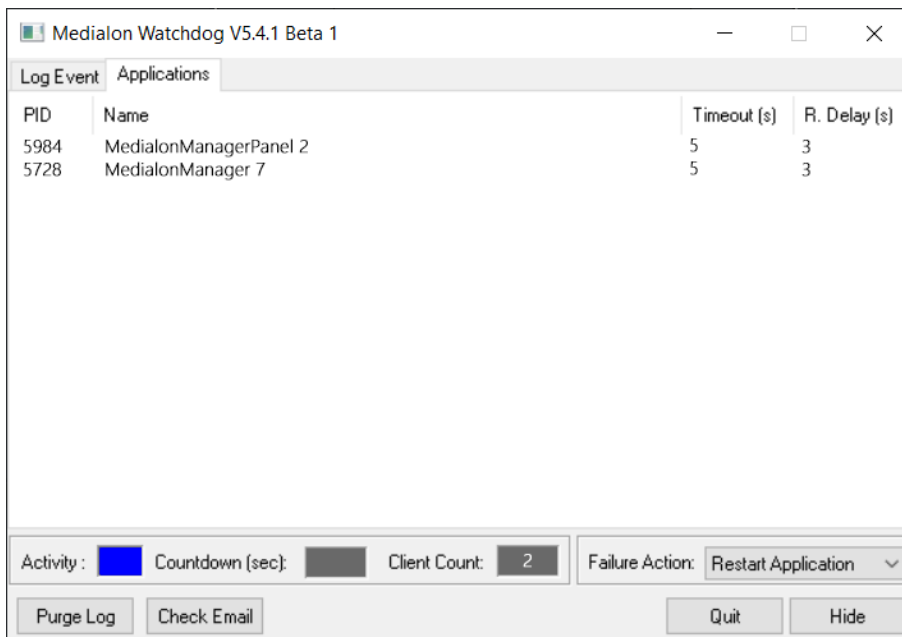
The Countdown starts when a Manager V7 failure is detected. Restart delay is set in Manager: File > Preferences > Watchdog.

Client Count

Displays how many applications are monitored by Medialon Watchdog.

Applications

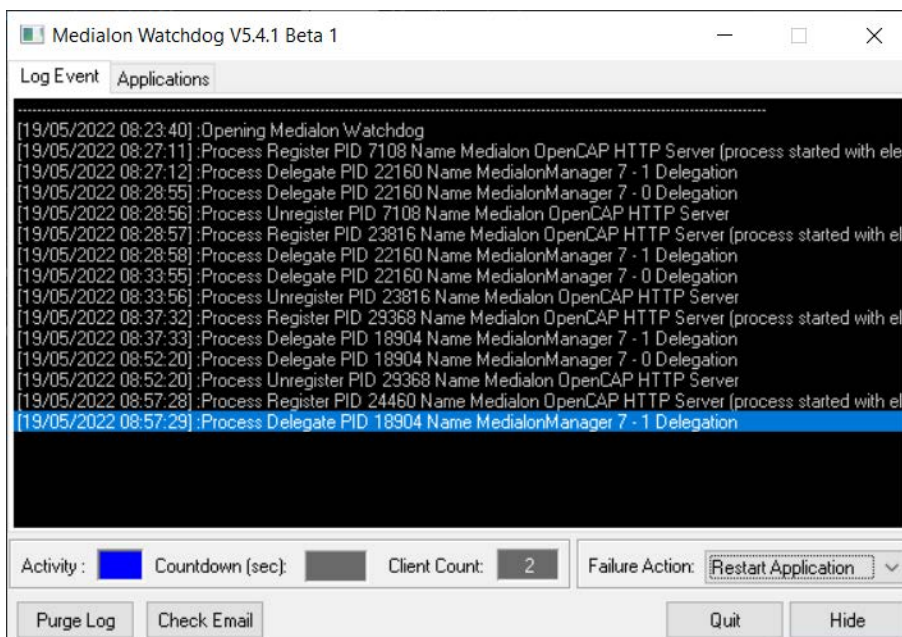
The Applications tab displays the list of applications monitored by Medialon Watchdog.



For each application PID (Process Identification number), timeout and restart delay are displayed.

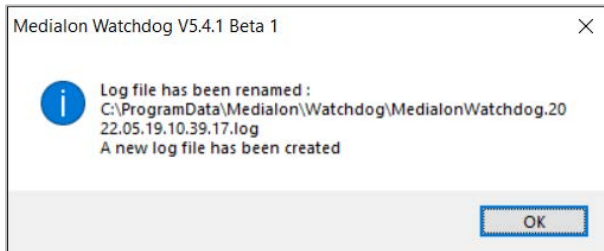
Log Event

The Log Event tab displays all actions of Medialon watchdog.



Purge Log

All events of Medialon Watchdog are record into a log file. The Purge Log button allows you to close the current log file and clear the LogEvent list. A new log file is automatically created.



Log files are saved by default in the C:\Windows\system32 directory. Edit 'MedialonWatchDog.cfg' file (in the Medialon Watchdog folder) with a text editor to change the file location.

Email Notification

Medialon Watchdog can send an email in case of failure detection.

Email settings can be specified inside the email.xml file in the Medialon Watchdog folder.



Click the Check Email button to send a test email to validate your settings.

9.4 Project Files

9.4.1 Project Files and Folders

A Medialon Manager V7 Project is composed of a project file and an associated folder. A variable storage file is also generated if the project uses persistent variables.

Medialon Manager Project File


A Medialon Manager V7 project is saved in a file with the extension `.mngProj` .

A Medialon Manager Panel project is saved in a file with the extension `.mngPanelProj` meaning Medialon Manager Panel Project File

Image Folder

If a project uses User Bitmap Images in UserScreens or in the Device Map, the bitmap files are saved in the `ProjectName.ImageFolder` located in the same directory as the project file.

Persistent Variables Files

The values of all of the persistent variables defined in a project are saved in a file with the same name but with the extension `.mngVarStorstor`  (Manager Variable Storage). This variable storage file is updated each time a persistent variable changes.

Another file with the extension `.mngTmpVarStor` (Manager Variable Temporary Storage) is created as a copy of the `.mngVarStor` file. If power to the computer is lost during a write operation to the `.mngVarStor` file, the data could become corrupted and values of all persistent variables would be lost. The `.mngTmpVarStor` file prevents this from happening, as it automatically backs up the `.mngVarStor` file as soon as the file is finished being written. If Manager detects a corruption of the `.mngVarStor` file, it uses the data from the `.mngTmpVarStor` file.




A project can be easily moved from one computer to another by a standard file copy operation. Note that the `.mngVarStor` and the `Project.Image` folder must be copied along with the project's `.mngProj` file in order to copy the entire project.

9.4.2 Recovering the Previous Version of a Project

Each time you save creates a new version of a project. A copy of the previous version is created in the same directory. When a project file is saved, two files are created or updated.

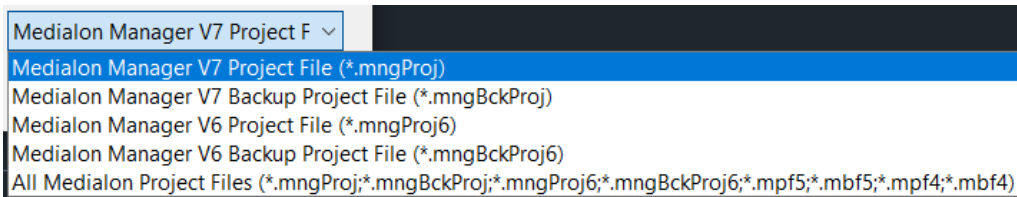
- Manager V7 Project File (`.mngProj`)
- Manager V7 Backup File (`.mngBckProj`)


The 'Medialon Manager Backup File' , appears following the second save under the same name. The current file's extension is '.mngProj' while the backup file is under the extension '.mngBckProj'.

The Backup File is a copy of the previously saved version of the project.

9.4.3 Opening Project from previous version of Medialon Manager

You can open a Medialon Manager Project created with a previous version of Medialon Manager:



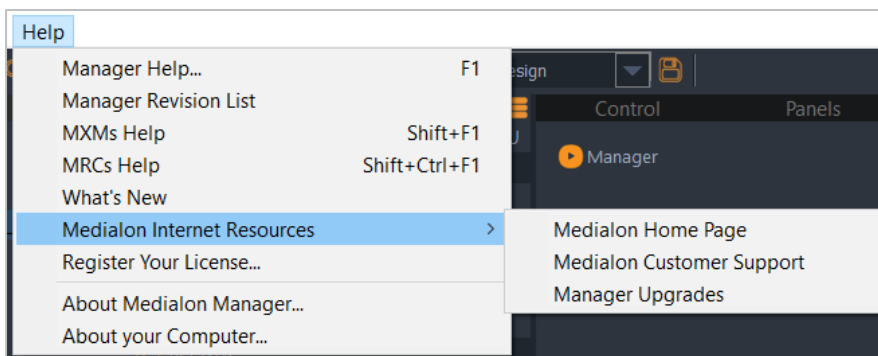
 Projects made with Medialon Manager V2 or 3 cannot be opened with version 7; you need to open them with version 4 or 5 first.

9.4.4 Upgrading Manager V7, MRCs and MxMs

Manager V7, MRCs and MxMs are regularly updated to correct minor problems or to add functionality and commands. It is therefore important, especially when starting a project, to have the latest version of the MRCs and MxMs.

Medialon Manager V7, MRCs and MxMs are never updated automatically, even if an internet connection is available. This operation must be performed manually by the system administrator.

To download the new versions of Manager V7, MRCs and MxMs, you can access the download pages through the Help menu, 'Medialon Internet Resources / Manager V7 and MxMs Upgrades' option. This will direct your default browser to the Medialon download pages.



You will have access to all recent versions of Medialon Manager and to the latest version of Medialon MRCs and MxMs.

Manager V7 Specific Features

The downloaded files are zipped .exe files to be launched on the Show Control system to be updated. Backup your project files and quit Manager V7. Unzip the downloaded file and double-click on the .exe file to perform the update.

ANNEXES

Annex A: Medialon Resource Connector

i Independently from MxMs, Medialon Manager uses resources to physically connect to devices.

i Medialon Resource Connector (MRC) manages and configures the necessary resources for your projects.

Embedded Resources

Medialon Resource Connector (MRC) manages and configures the necessary resources for your projects.

All resources installed in your PC or Showmaster are scanned when Medialon Manager starts and are automatically mounted in your project.

MRC Windows COM

All Windows COM ports installed in your PC are scanned when Medialon Manager V7 starts. MRC Windows COM gives access to Serial Ports.

MRC Windows MIDI

All Windows MIDI ports installed in your PC are scanned when Medialon Manager V7 started. MRC Windows MIDI gives access to MIDI Ports

External Resources

Medialon Resource Connector (MRC) manages and configures the necessary resources for your projects.

Medialon Manager can manage external resources, like networked resources as Art-Net, Advantech, AMX NetLinx, Global Caché, etc.

MRC AMX NetLinx

AMX is a manufacturer of Remote Control systems offering a wide range of controllers, keypads, and touch panels.

With smart processors and powerful memory reserves, AMX NetLinx® Controllers can simultaneously carry out multiple commands to produce instant, flawless results from the devices they control.

When connected to a Medialon Manager V7 system via Ethernet, AMX NetLinx® Controllers offer their hardware resources to Medialon Manager Software. There is no need for programming in the AMX NetLinx® itself. Medialon Manager V7 remains the main controller and sends commands from its own MxMs and drivers.

The combination of the two systems, Medialon Manager V7 Show Control software and AMX highly reliable hardware, creates a great solution for many applications where timelines, accurate synchronization, logical operations, database links, digital media control, wide network deployment, etc. are necessary.

MRC AMX NetLinx gives access to:

- Serial Ports
- MIDI Ports
- I/O Ports
- Infrared Ports

MRC Global Caché

Global Caché is leading the way in automation with hardware solutions aimed at connecting a network to the devices that we interact with every day.

The GC-100 Network Adapter, iTach modules and iTach Flex modules connect TCP/IP networks to infrared (IR), serial, relay outputs and sensor inputs. An embedded web server allows easy configuration from any browser.



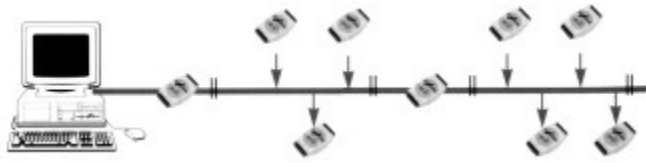
MRC Global Caché gives access to:

- Serial Ports
- I/O Ports
- Infrared Ports

MRC Adlink NuDAM

ADLINK's Intelligent Remote Data Acquisition and Control Modules (NuDAM) are designed for data acquisition systems based on PCs and other processor-based equipment with standard

serial I/O ports (RS-232 or RS-485 with auto-direction control). The modules convert input/output signals to engineering units and transmit/receive, in ASCII format, to/from any host computer with RS-232 or RS- 485 port.

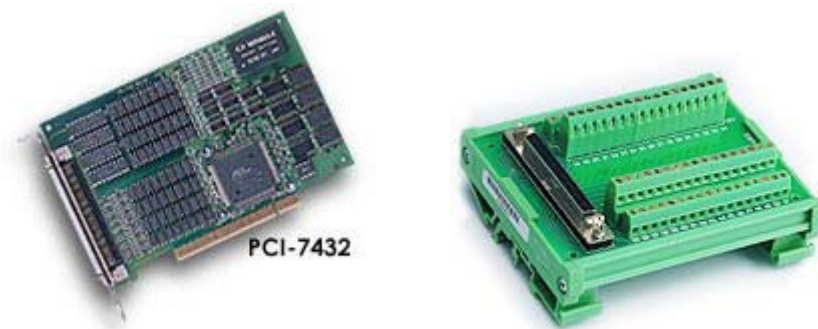


MRC ADLINK NuDAM gives access to I/O Ports.

MRC ADLINK PCI-7432

The ADLINK PCI-7432 board is a 32 isolated digital input and 32 isolated digital output (open collector) channels board. Each output can sink 5 V DC to 32 V DC loads with a maximum of 500 mA in total. Each input can handle from 0-24 V DC with a resistance of 2.4 k Ω @0.5 W.

The isolation voltage is up to 5000 V rms.



DIN-100S terminal board can be connected to the card using a SCSI 100 Pin cable. MRC ADLINK PCI-7432 gives access to I/O Ports.

MRC Adrienne

Adrienne Electronics provides the Adrienne RG1, a timecode LTC reader and generator board that fits into a PCI slot in a regular PC.



Annex A: Medialon Resource Connector

It can either sync to video or free run with very high precision. The AD RG1 can read and generate LTC simultaneously.

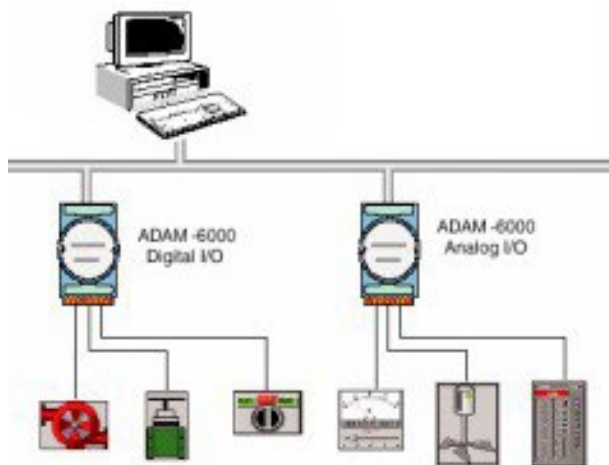
The Adrienne USB-TC is a timecode reader USB. These small but rugged metal boxes are perfect for laptop and other slot-limited applications and are both USB 1.1 and 2.0 compliant.



MRC Adrienne gives access to Timecode Ports.

MRC Advantech ADAM

Advantech ADAM-6000 Ethernet-based data acquisition and control modules provide I/O, data acquisition and networking in one module to build distributed monitoring, and control solutions for a wide variety of applications. Through standard Ethernet networking, ADAM-6000 receives I/O values from sensors and publishes these real-time I/O values.



MRC Advantech ADAM gives access to I/O Ports.

MRC Art-Net

Art-Net is a communication system that allows DMX512 to be sent over an Ethernet network. It was invented by Artistic Licence and is now published in the public domain.

MRC Art-Net gives access to DMX Ports.

MRC Control DeviceMaster

The Control DeviceMaster series are Serial port servers communicating over Ethernet. This MRC uses the TCP/IP Telnet Control Port protocol (also referenced as RFC-2217) to communicate with DeviceMaster servers.



MRC Control DeviceMaster gives access to Serial Ports.

MRC ICP DAS PISO-P32x32

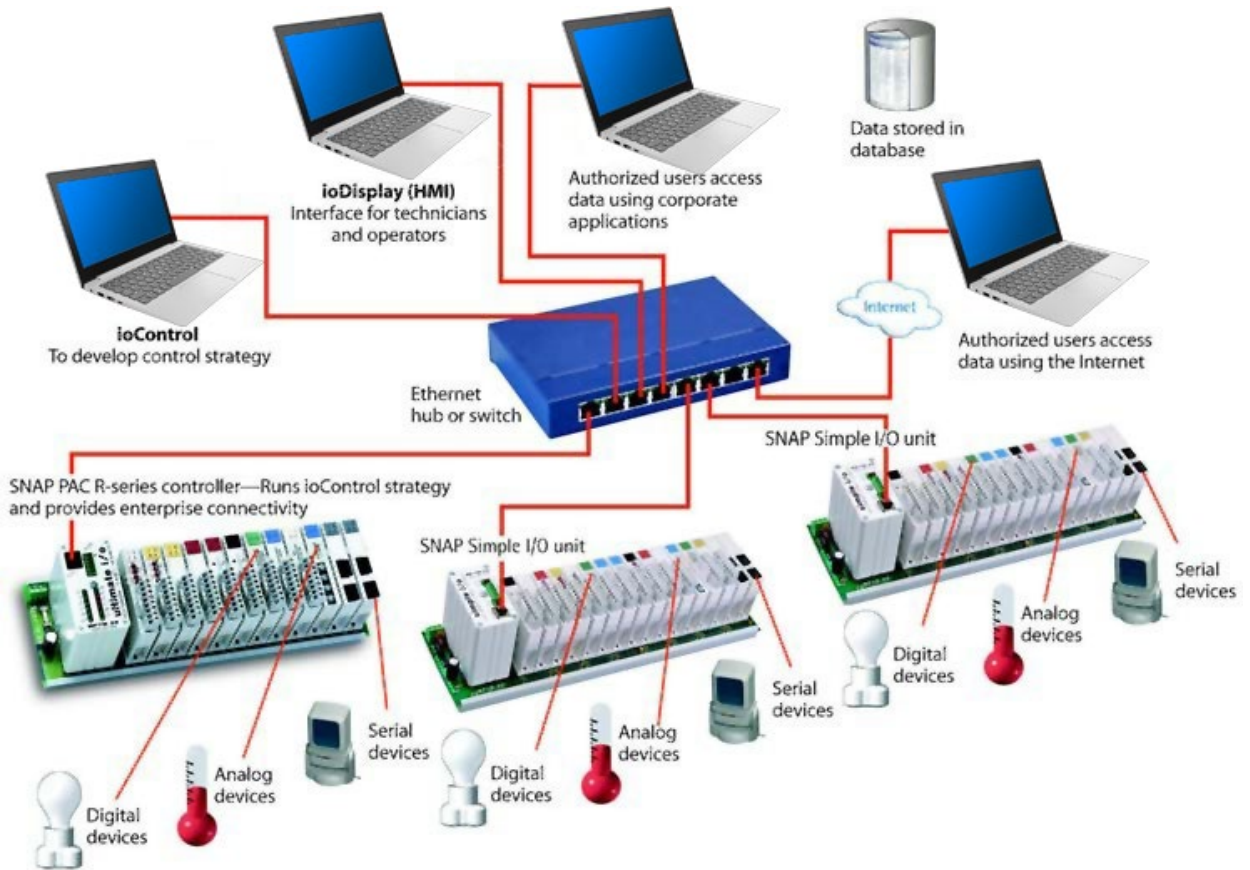
The PISO-P32C32 and PISO-P32A32 have 32 channels of optically isolated digital inputs and 32 channels of optically isolated digital outputs, arranged into four isolated banks. Each input channel uses a photo-coupler input which allows either internal isolated power supply or external power selected by jumper. Each digital output offers a Darlington (P32C32) or PNP (P32A32) transistor and integral suppression diode for inductive loads. Isolated input channels 0-15 are designed into group A and channels 16-31 are designed into group B. Isolated output channels are designed into group C and group D.



Wiring can be done using two DN-37 terminal board, one for each group of 16 channels. MRC ICP DAS PISO-P32x32 gives access to I/O Ports.

MRC Opto22 SNAP Ethernet IO

Opto22 SNAP Ethernet I/O is an Input/Output connection module. Using Opto22 SNAP Ethernet I/O systems, you can connect a wide variety of electronic and mechanical devices, such as lights, temperature and pressure sensors, motors, serial devices, and more to computers using Ethernet networks, wireless LAN networks, and even the Internet.



i MRC Opto22 SNAP Ethernet IO only supports Digital Input modules and Digital Output modules.

Opto22 SNAP digital I/O modules handle devices that can be in only one of two states: either on or off (true or false, 0 or 1, running or stopped, etc.). Push buttons and LEDs are examples of digital devices. SNAP digital input modules send data from the device to the computer, translating signals from the device into a form that computers can understand, while maintaining a protective barrier between the computer and the dangerous high voltages or other hazards of the real world.

MRC Opto22 SNAP Ethernet IO gives access to I/O Ports.

MRC Sealevel SeaLINK

The Sealevel SeaLINK series are Serial ports servers communicating over Ethernet. This MRC uses the TCP/IP Telnet Control Port protocol (also referenced as RFC-2217) to communicate with SeaLINK servers.





MRC Sealevel SeaLINK gives access to Serial Ports.

MRC SoundLight

The SoundLight DMX 1514PCI is a versatile DMX-512 interface card that fits into any PC running Medialon Manager V7 with a PCI Slot.



The SoundLight DMX PCI offers up to 1024 DMX input channels or 1024 DMX output channels or combination (512 inputs and 512 outputs). This card handles two complete DMX in/out links in parallel. Up to four cards can be installed in one PC. Soundlight cards are no longer in production, but the MRC is still available to support legacy installations.

MRC SoundLight gives access to DMX Ports.

Annex B: Medialon MxMs

Medialon Manager sends commands to external devices via software modules called *Medialon XObject Modules (MxMs)*.

Service MxMs

Some MxMs implement services as extensions of Medialon Manager and provide sophisticated functionality like scheduling, error logging or database management... They are referred to as 'Service MxMs'. These MxMs greatly enhance Medialon Manager and provide the necessary resources to manage and control information or perform special tasks.

Data Management

MxM Database

This MxM allows interfacing of SQL compatible databases. It allows reading, modification and creation of data and records, and sending of SQL requests to the database. It interfaces with various database types using the ODBC data sources configured in Windows.

MxM Files Management

This MxM provides a simple way to copy, move or erase files on the hard disk. It also provides commands for reading and writing files on the local disk or on any accessible shared directory over the network. It can also be used to monitor changes in a folder to trigger actions.

Special commands are provided for managing INI files. INI files are very practical for creating simple database files. They contain sections and items.

For example: a simple INI file called 'Addresses.INI' where names of people are referred to as sections and their addresses are associated items.

MxM FTP Client

MxM FTP Client is designed to provide File Transfer Protocol Client facilities. It provides functions to connect to, download from, upload to, and manage FTP Sites from regular FTP servers (MxM FTP Client is compatible with at least Microsoft IIS FTP Server, Bison FTP Server and CatSoft Serv-U server).

Multiple and filtered file transfers can be done with a single command like download '*.txt' or Upload 'DIR' where 'DIR' is a directory name.

MxM Strings Management

This MxM provides conversion and analysis of character strings. A series of commands is provided for manipulating strings, for the purpose of saving and sorting, arrangement of

strings in blocks, and refining of strings from blocks. For example, these commands are practical for managing and saving tables of ASCII text strings.

Utilities

MxM Medialon AppLauncher

This MxM provides a way to control Windows PCs and third-party software from Medialon Manager. The application to be controlled can reside on the local computer or on a remote computer.

If it resides on a remote computer, a tool called Medialon AppRemote runs on the remote computer and is controlled from a Medialon Manager using MxM Medialon AppLauncher. The MxM provides a wide range of commands including Start Program, Stop Program, Windows Size, Bring to Top, Shutdown PC, etc. to fully control software behavior. It is also possible to target specific windows and send keypresses using the Key command, simulating local keyboard strokes in the target application.

A command line can be added to the Start Program command. Knowledge of the application to be controlled (command line documentation, shortcuts, menus...) is recommended.

MxM EventLog

MxM EventLog is designed to store custom logged events and produce event reports. It provides functions to store, sort, export and print event lists by date and time.

All traced events are stored in the current log file which is part of the log database of events. The events storage file can be changed automatically every period which can be defined as every day, week, month, or year.

MxM HTTP Remote

This MxM provides a rudimentary interface between a Medialon Manager project and a web browser. The MxM communicates with the web browser via the HTTP protocol and the exchange of data is done in XML syntax.

The MxM provides functions for creating controls in an HTML page. These controls (buttons, lists, check boxes, edit controls...) are designed for communicating with the Medialon Manager project. They can be added to any existing HTML page, or created in a new page, from the setup window of the device.

The display properties of the controls are basic, but the page can be easily styled with an HTML editor.

Medialon Manager variables can be read and written from the HTML page. On Medialon Manager's side, the MxM can be easily imported into a project. *But on the browser side, a knowledge of HTML and JavaScript is necessary* if special mechanisms are to be implemented (for simple controls, the pages exported by the MxM are sufficient).

This MxM is provided for legacy compatibility. WebPanels are the preferred method for interfacing with web browsers in the current software version.

MxM Network Utilities

The Network Utilities MxM provides several commands for network management and retrieval of system information.

MxM Pictures

Pictures MxM provides a way to dynamically display still pictures and editable text in Medialon Manager UserScreens. Pictures supported format are JPEG, BMP, ICO and WMF. Images can be stretched and scrolled at adjustable speeds, text can be written and scroll in the same way. The user can easily create a task in Medialon Manager for displaying sequences of images.

MxM Variables Management

This MxM provides commands that act on variables, and these commands are Positracked when they are used in a timeline. It is possible to create a counter or timer with this MxM.

Telecom

MxM Email

This MxM offers the possibility to manage the sending and receiving of emails in a Medialon Manager project. It has the basic functionality of an email software, including management of modem connections.

MxM Telephony

This MxM provides controls for telephony resources present in the system. These telephony resources are usually modems or virtual modems created by communication software.

The drivers of these telephony resources must be compliant with the Microsoft Win32 Telephony application programming interface (marked as 'TAPI compliant').

To use DTMF monitoring, the resource must be at least compliant with the version 2.0 of TAPI.

Device MxMs

MxMs that allow communication with external devices are referred to as 'Device MxMs'. Device MxMs allow Medialon Manager to communicate with hardware machines such as video projectors and lighting consoles, but they can also control third-party software such as Dataton Watchout. A wide range of Device MxMs are available for your show control application.

Driver Editor

MxM Medialon Low Level Communicator

The Low Level Communicator MxM is a low-level and powerful MxM. It is a low level MxM because it is not dedicated to a particular device and is able to use many raw communication protocols.

You can write your own set of commands, create monitoring variables linked to receiving specified frames, and much more, to create drivers dedicated to your specific devices and projects.

Five type of communication are available:

- Serial
- TCP/IP Client
- TCP/IP Server
- UDP/IP
- MIDI

MIDI

MxM Medialon MIDI In/Out

This MxM offers basic control of the MIDI interface. It can be used to control any MIDI compatible devices, such as audio mixing desks, lighting consoles, audio sequencers, fireworks controllers, etc.

Each device can be assigned to control one MIDI in port and one MIDI out port. Two devices can share the same pair of ports (one device for Input, one for Output).

MxM Medialon MIDI AudioController

This MxM provides MIDI controls for audio boards. It assumes that the audio sliders are controlled by the 'control change'. It also includes commands for 'Program Change', 'Note On' and 'Note Off'.

MxM Medialon MIDI Machine Control

This MxM provides the most common MIDI Machine Control commands. These commands are setup to control a VTR device with a MIDI port. Both the MIDI output and input ports should be used to provide full duplex dialog between the MxM and the controlled device.

MxM Medialon MIDI Show Control

This MxM provides the most common MIDI Show Control commands. These commands are set up to control dedicated intelligent control equipment in theatrical, live performance, multimedia, audiovisual and similar environments.

Applications may range from a single interface through which a single lighting controller can be instructed to GO, STOP or RESUME, to complex communications with large timed and synchronized systems utilizing many controllers of all types of performance technology.

This MxM provides the capability to receive and react to MSC commands to perform programmed actions.

Lighting

MxM Medialon DMX512

This MxM provides commands to control groups of DMX512 channels through several DMX512 hardware devices.

Inputs and/or Outputs can be controlled. Input sensing can analyze incoming DMX values; commands are available for live recording and restoring of the DMX-512 signal.

This MxM can also create and play back memories and save channel values in an external file.

This MxM includes a graphical cue editor interface. This interface allows the programmer to create, edit, and retrieve cues on selected channels. A graphical track ball allows programming of several channels for X and Y to ease control of automated lighting fixtures.

It is possible to record live DMX coming from a lighting desk, and then edit the cue snapshots which were created as DMX cues with a user-friendly graphical interface.

Timecode

MxM Medialon Timecode

This MxMs allows reading and generating of Timecode using Timecode resources.

Relay I/O

MxM Medialon IO

This MxM controls and monitors various types of Inputs/Outputs (I/Os) with a unified programming and user interface.

MxM Leunig ePowerSwitch

The Leunig ePowerSwitch is an intelligent power strip with its own web server and an Ethernet connection. With ePowerSwitch, you can switch devices on and off through a local network or the Internet. The MxM provides control of the ePowerSwitch.

Field Bus

MxM Medialon Modbus TCP

The Medialon Modbus TCP MxM give access to devices that support Modbus TCP protocol; it acts as a Modbus TCP/IP Client.

Values in registers (or inputs or coils) and register organization is device dependent.

The user should know how the device is organized and what the available registers (or inputs or coils) are in the device.

More information about Modbus TCP can be found at www.modbus.org

MxM Medialon OPC Client

OPC – OLE for Process Control – was created as a vendor-independent standard aimed at the definition of data exchange between Microsoft Windows applications as well as applications and vendor-specific hardware in the realm of automation and industry. Many industrial hardware vendors provide OPC servers for their products like data acquisition boards, PLCs, fieldbus controllers, etc.

OPC defines a series of standard specifications. This MxM acts as a client for both 'OPC Data Access' servers and/or 'OPC Alarms and Events' servers.

MxM CIP Ethernet IP

This MxM allows Manager to interface with a Rockwell Automation Control Logix family PLC using the CIP Ethernet/IP protocol to read and write symbolic references created in the PLC.

This MxM supports the basic PLC integer types with size 8, 16, or 32-bit.

Infrared

MxM Medialon Infrared

This MxM controls various types of Infrared outputs with a unified programming and user interface.

The infrared resources themselves are provided by MRCs (Medialon Resource Connectors) that implement Infrared Resource management, like 'Global Cache' MRC.

Audio Processor

MxM BSS Soundweb

This MxM provides serial remote control to BSS Soundweb devices.

Soundweb is a set of audio processing units that can be linked by a digital communications network. The units are completely flexible; you program the signal paths and block diagram of the processing using a PC, choosing the processing blocks from an extensive library. See also www.bss.co.uk for more details).

MxM BSS Soundweb London

This MxM controls and monitors several types of Soundweb London. Each MxM instance can control one London device, either via a serial connection or via Ethernet (TCP/IP).

The hardware configuration and internal patch creation, loading and saving are done by using the BSS London Architect PC application. This MxM does not provide for building DSP graphs from scratch and is not intended to replace or supersede the London Architect software.

By using this MxM, one can control and monitor the state of any built-in or custom DSP object inside the London's internal DSP graph.

Built-in objects correspond to the common controls found in each London unit (like input gains and meters) and are created automatically when completing the MxM device setup. User objects can be added in the device setup as well.

MxM Merging Ovation

This MxM provides control over the Merging Ovation audio/video server. It allows Ovation to be used in an integrated show control environment and provides simple cue list and cue firing capabilities as well as a Positrack model which simplifies programming of mutually exclusive cues, allowing shows to be edited in small chunks.

The Ovation Positrack model supports a single active cue per group with time-based tracking of that cue. This allows the programmer to navigate through a timeline sequence and restart the show with the Ovation replaying the correct cue at the correct point in time. An unlimited number of groups can be created which are not tied to specific tracks in the timeline.

MxM Panasonic WR-DA7

This MxM provides an easy way to control one or several Panasonic WR-DA7 audio mixers from the MIDI output.

MxM Peavey MediaMatrix

This MxM provides IP remote control of the Peavey MediaMatrix software and NiON units using the RATC-1 protocol.

MediaMatrix is a versatile digital audio resource that allows you to design and build audio control systems on a function-built computer equipped with specialized audio hardware and the MediaMatrix control software.

MxM Peavey MediaMatrix Pasha

This MxM provides serial remote control to the Peavey MediaMatrix software. The PASHA interface must be enabled in the MediaMatrix software.

MediaMatrix is a versatile digital audio resource that allows you to design and build audio control systems on a function-built computer equipped with specialized audio hardware and the MediaMatrix control software.

MxM QSC Audio Q-Sys

Q-Sys is an Integrated System Platform that provides all the audio routing, processing control and monitoring for a facility. Q-Sys is a centralized DSP, integrated-system audio solution.

This MxM can control Q-Sys cores: core control values monitoring through variables, control values sending through commands, and redundancy management between two cores.

MxM Richmond SoundMan

This MxM provides IP control of the Richmond SoundMan Server.

SoundMan Server is a virtual sound system. It is an audio routing, playback and processing solution that runs under Windows and uses ASIO drivers. The software runs as a server and thus can be controlled externally by TCP/IP on the network.

MxM RMS Sinfonia

This MxM provides an easy way to interface the Sinfonia system from Real-time Music Solutions. Its patented technology allows for easy programming of very subtle and nuanced performances. Treated like a musical instrument, Sinfonia® becomes transparent in the pit. The traditional ensemble never plays to a click track. Sinfonia®, as played by a musician, follows the conductor. It becomes a natural part of the ensemble with an integrated sound quality that matches the acoustic instruments.

Audio Server

MxM Medialon Audio Server

Medialon Audio Servers are audio servers that can control up to 128 independent audio channels. Each channel can be loaded with a media file (WAV) and then played, paused, stopped, or located independently. This MxM controls Medialon Audio Servers via TCP/IP on an Ethernet network.

Camera

MxM Sony Camera VISCA

VISCA is the protocol used by Sony for remote control of the BRC-300 color video camera series. This MxM provides control of the video camera as a simple remote control unit, but also includes a powerful sequence editor.

Matrix

MxM Extron 100

The Extron 100 MxM provides an easy way to control one or several Extron 100 matrix switchers from an RS-232 Serial port.

MxM Extron 3200/6400/12800

The Extron 3200/6400/12800 MxM provides an easy way to control one or several Extron 3200/6400/12800 matrix switchers from an RS-232 Serial port.

This MxM also supports CP450+ and MAW Matrix models.

MxM Pelco CM9760 Matrix Switcher

This MxM controls the Pelco-9750 and Pelco-9760 Video Matrix series.

MxM Philips LTC 8300

This MxM controls the Philips LTC 8300 series video matrix.

Because only the basic functions of this matrix are treated, this MxM is also compliant with the LTC 8500, LTC 8600, LTC 8700, LTC 8800 and LTC 8900 series.

MxM ProBel Matrix Switcher

The ProBel MxM provides an easy way to control one or several ProBel matrix switchers, from an RS-232 Serial port.

MxM Rouvroy Matrix Switcher

The range of Rouvroy switching matrixes offers several interesting features which make them particularly powerful in automated installations. They are available with composite video, audio, RGB, Serial signal, Audio with volume control etc. All combinations are possible on request.

This MxM offers all switching and volume control commands. It also offers management of inputs and output labels and a memory module that makes it possible to create and use switching memories that can be stored in a file. This file can be reloaded or used by several Medialon Manager systems.

Rigging

MxM Stagetech Rigging System

The Stagetech Rigging System (SRS) is a software implemented in PLCs such as the Simotion from Siemens by the Stagetech company.

This MxM can control multiple SRS through Ethernet. The MxM controls scenic elements on the SRS; these elements are called 'axis'. The MxM also controls output pins on the SRS (i.e. relay-like outputs) and monitors input pins.

Video Processor

MxM Analog Way Graphic Switcher

This MxM provides serial remote control to the Analog Way Graphic Switcher.

The seamless GRAPHIC SWITCHER II™ inserts (PIP), cuts, fades, wipes and instantaneously mixes (with no glitches) between 16 high-resolution and TV/ Video sources, which can range from 15 kHz to 130 kHz (Auto Sync and Auto Scan), up to 1600 × 1280, with no synchronization 'dropouts'.

MxM Analog Way IXSeries

Analog Way IXseries products provide Hi-Resolution Mixers and Seamless Switcher and offer a new high-quality scaling processor with MATRIX and MIXER modes.

The MxM is used to create device instances inside Medialon Manager. A device represents a processing system as a whole and can 'talk' to the underlying hardware. Each device can control between 1 and 16 processors.

Each processor represents a single hardware mixer or matrix.

MxM Analog Way OctoPlus

This MxM provides serial remote control of the Analog Way Octo-Series devices.

The OCTO-Series is a computer and video up/down scaler switcher with 8 universal A/V inputs. It performs fast and clean transitions between any scaled video and computer sources. It also allows true seamless switching between one computer input (direct) and any other scaled video or computer input.

MxM Barco Folsom Encore

This MxM provides remote control of the Barco Folsom Encore system.

MxM GrassValley Encore

The Encore™ system is an open, scalable platform for full router and facility control that features tight integration with automation systems, third-party tally and UMD systems, and other equipment. This scalable system gives everything from simple crosspoint configuration to real-time centralized and distributed machine control.

With the help of the MxM, you will be able to take (commute) one source to one destination on a specific level (layer). You will be able also to take, in one shot, multiple sources to one or more destination on specific levels. The connection to the Encore™ is done via Ethernet (TCP/IP).

Video Projector

MxM Medialon PJLink

PJLink is a standard protocol for controlling projectors on network defining the following: Procedures for connection to projector via network, Security, Control command form.

Medialon PJLink MxM supports the Class 1 which performs the fundamental control of projectors of different manufacturers or models.

MxM Barco CRT

The BARCO CRT MxM provides an easy way to control one or several BARCO CRT video projectors from Medialon Manager's RS-232 Serial ports. Several Barco projectors can be controlled independently from one Serial port, using different addresses on each video projector.

MxM Barco DLP

The BARCO DLP MxM provides an easy way to control one or several BARCO DLP video projectors from Medialon Manager's RS-232 Serial ports. Several Barco projectors can be controlled independently from one Serial port, using different addresses on each video projector.

MxM Barco FLM CLM Projectors

The BARCO FLM CLM MxM provides an easy way to control BARCO FLM or CLM video projectors via Ethernet (TCP/IP) or RS-232 / RS-422 Serial ports.

MxM Barco LCD

The BARCO LCD MxM provides an easy way to control one or several BARCO LCD video projectors from Medialon Manager's RS-232 Serial ports. Several Barco projectors can be controlled independently from one Serial port, using different addresses for each video projector.

MxM Christie Projectors

The Christie Projectors MxM provides an easy way to control one or several video projectors manufactured by Christie through a Serial line or via a TCP/IP network.

The types of projectors driven by the MxM include the Vista, Roadster, Mirage and Roadie series.

MxM Digital Projection Projectors

The Digital Projection Projectors MxM provides an easy way to control one or several video projectors from Digital Projection LTD through a Serial line.

The types of projectors driven by the MxM include the MT series, LT series, SX series and GT series.

MxM Sony VPL Network Projectors

The Sony Network Projectors MxM provides an easy way to control Sony video projectors that implement PJ-Talk via TCP/IP.

Video Server

MxM Medialon Local Media Player

This MxM provides a simple and fast way to reach and play various type of media from your PC hard disk or from a distant PC or server hard disk on the network. The media can be video or audio, in various formats.

Video files can be played in Full screen or windowed mode, and many media files can be played at the same time (depending on your PC configuration). This MxM contains all useful VTR functions as Play, Pause, Stop, and Locate.

This MxM makes it easy to create an Mpeg video player. To play an Mpeg2 video, an Mpeg2 decoding circuit board or software decoder is necessary, such as Sigma Design Netstream 2 or Moonlight Elecard MPEG2 Decoder.

Video can be both displayed in a Medialon Manager graphical object container and sent to a video projector using the Mpeg2 decoding circuit board's analog output. Video can also be displayed outside an object container. In this case, settings of video window aspect should be done in the setup dialog box.

This MxM is based on the Microsoft DirectShow Layer and the supported formats depend on the DirectShow filters installed on your PC. Most common are MPEG 1 and 2, WM9, AVI, Mp3, MIDI, WAV, Seq. These filters can be input filters like the UDP reader filter or decoder filters like the MPEG2 Decoder filter.

In some cases, a Hardware Mpeg 2 decoding circuit board for Mpeg2 files is required to perform fast Mpeg 2 decoding.

This MxM can synchronize Video playback to an external reference.

This reference is provided via TCP/IP by another MxM called MxM Medialon SyncMaster.

MxM ADTEC MPEG Player

ADTEC MPEG Players are hard drive based digital video players and network appliances designed for integration into global multimedia networks. ADTEC units can be linked via Ethernet and can play synchronously (for Video Walls and Multiple Projector applications). This MxM is compatible with the Edge (up to the 4013 model), Soloist and Mirage models. Note that this MxM is not compatible with Edge HD 4111 (but there is a Low Level Communicator driver for it).

MxM Alcorn McBride DVM

The Digital Video Machine is a studio-quality MPEG-2 player with all the interfaces AV contractors need for professional installations. The Digital Video Machine stores its videos on Compact Flash cards, so playback is virtually instantaneous. You can copy clips to the media or transfer them using the Ethernet option.

It can be controlled either using the serial port or using Ethernet (Ethernet requires the DVM-7400/E model).

MxM Dataton Watchout IP

Watchout is a scalable system for presentations based on digital media. This software allows you to create big shows while maintaining unprecedented image fidelity. Images can span across several video projectors or monitors, if desired. Watchout automatically applies smooth edge blending to overlapping projection areas, resulting in a seamless image. You can apply transitions to images or image elements, as well as animate them across screen areas. Video can be integrated and used in the same way. You can import a wide variety of video and image formats, making it compatible with virtually any production software you may already have. It can be used whenever you need to display or project high quality digital media. While it can be used for single screen applications, it is primarily designed for multi-screen use.

MxM Doremi Labs V1 and Nuggets

Doremi Labs offers a range of MJPEG, MPEG and uncompressed high quality video players and recorders. This MxM controls Doremi Labs V1 or Doremi Labs Nugget machines. In addition to standard VTR commands, this MxM provides Clip Management functions.

The connection between the MxM and the device uses either a RS422 connection (Sony 9 Pin connector) or a Network connection (UDP/IP).

MxM DVS Blade

The Blade is a small solid-state controlled playback system that delivers broadcast-quality video and audio.

This MxM controls and monitors the Blade (plays videos, executes scripts, controls volume, etc.).

MxM Ektacom Streamcoder

Ektacom Streamcoders are network video streaming servers. They can:

- Encode video streams in either MPEG1, MPEG2 or WindowsMedia9 format.
- Stream the encoded videos over the network in HTTP Unicast or RTP/UDP Multicast.
- Record the video into files.

Ektacom Streamcoders provide a set of 'profiles'. Each 'Profile' specifies a great number of capture and encoding parameters that can be simply recalled from either this MxM or the Ektacom WEB administration tool.

MxM GrassValley K2

The Grass Valley K2 is video server which supports 4 playback/record streams.

This MxM controls a Grass Valley K2 with the AMP protocol either via Ethernet or RS-422.

MxM GrassValley Turbo

The Grass Valley Turbo iDDR is video server that supports 2 playback streams and 1 record stream. This MxM controls a Grass Valley Turbo with the AMP protocol either via Ethernet or RS-422.

MxM Iridas FrameCycler

Iridas FrameCycler is a software product with several editions: FrameCycler Professional, DDS, DI, and Venue System. All these products can be controlled using the Iridas FrameCycler MxM. Basic operations are directly available such as 'Open Sequence', 'Play', 'Pause', etc. Advanced control can be performed using the generic 'Dispatch Command', which can send virtually any kind of command to FrameCycler.

MxM Louth VDCP

The Louth VDCP protocol is a popular RS-422 control protocol for Video Disc Recorders or Video Servers. This MxM implements the main functionality of VDCP.

With this MxM, the user can build a PlayList of clips by using 'Load Clip' / 'Load Next Clip' commands and then let the MxM play the playlist seamlessly.

MxM Medialon SyncMaster

This MxM can be used to synchronize Video playback on several PCs running the Medialon Local Media Player MxM.

MxM Odetics RS-422

The Odetics RS-422 protocol is a generic control protocol for Video Disc Recorders. It is a superset of the popular Sony RS-422 VTR control protocol. Most of the added functionalities are about 'Clip' management by which the user can queue 'Load Clip' / 'Load Next Clip' commands and then let the MxM play the clip playlist seamlessly. This MxM offers commands for all Odetics RS-422 compatible devices. Connection is made via a standard Sony 9 pin RS-422 cable.

MxM Sony RS-422

The Sony RS-422 protocol is one of the most popular VTR control protocols. You can find the protocol on most Sony Betacam and DV video recorders, and on some other manufacturers' devices. This MxM offers commands for all Sony RS-422 compatible devices. Connection is made via a standard Sony 9 pin RS-422 cable.

Annex C: Regional and Language Options

This section discusses the installation and configuration requirements and explains how to type in East Asian languages in Medialon Manager on computers that are running Microsoft Windows 10 or newer.

To configure the operating system to type in the appropriate language, you must configure the keyboard to type in that language. If the keyboard that you want is not listed by default, you may have to install additional files.

Configure Windows to type in multiple languages

Before you can type in a language other than English, you must configure the keyboard to match the language. For example, to type Chinese-based characters, configure the keyboard to map the same way as a localized Chinese keyboard. To do this, follow these steps.

1. Type 'Language' in Windows search to find Language Settings.
2. Click the + icon to add a preferred language and select the language you wish to install.
3. Install the language pack, but you may not need to set the display language of the PC. Click Install.
4. The language pack is installed. You can close the settings window.
5. In the bottom right corner of the display, near the system clock is a menu item for changing the keyboard input language (or use Win+Space). Switch to the keyboard layout of the target language.
6. As you type in text in Medialon Manager, an assistive input menu appears to assist with Asian language inputs.

Contact Us

Find more on our website www.7thsense.one and on social media.



or contact our support team at support@7thsense.one.

medialon

BY 7THSENSE

E: info@7thsense.one

W: 7thsense.one

7thSense Design Ltd

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

T: +44 (0) 1903 812299

7thSense LLC, Michigan

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

T: +1 248 599 2717

7thSense LLC, Orlando

4207 Vineland Rd
Suite M1
Orlando, FL 32811
USA

T: +1 407 505 5200
