Unison Mosaic Designer

User Manual

v1.11

Production information and specifications subject to change.

Part Number: **7180M1240-1.11** Rev A.

Released 2013-12.

Contents

Welcome

Introduction

Welcome and thank you for using version v1.11 of the Unison Mosaic Designer software. This release offers some significant improvements and additional functionality over earlier versions, see what's new for details.

WARNING: Projects saved with v1.11 can not be opened with earlier versions so please make sure to back up your programming prior to installation.

Platforms

Unison Mosaic Designer is designed to run on a Microsoft Windows 7, Vista or XP (SP2+) or Apple Mac OS X (10.4.x or later - Intel or PowerPC) personal computer. As the application is of a highly graphical nature a large monitor is recommended (1024x768 minimum) and the extensive use of OpenGL graphics routines will benefit from a high performance graphics card. USB or Ethernet support will be required to connect to the Unison Mosaic hardware.

Apple QuickTime

Apple QuickTime (6.5 or later) must be installed to run Designer and QuickTime 7 is supplied on the Unison Mosaic Designer installation CD. QuickTime is also available for download at Apple - QuickTime Player.

Help Overview

The Help is split into four major sections: Quick Start, Reference, Troubleshooting, and Appendices.

Those of you experimenting with the software for the first time should work through the Quick Start guide to familiarise yourself with the basics of the software. The Reference section then gives detailed descriptions of every aspect of the software as well as the configuration of the Unison Mosaic Controllers and their accessories. The Troubleshooting section provides help to resolve any problems while the Appendices provide additional useful resources.

If you have a Controller that you wish to connect to and program now then please read the Network section for instructions although this is not required when working through the Quick Start guide.

Help Help

This is the PDF version of the on-line Help and it is available in various formats for printing. The on-line version, which has the advantage of being fully searchable and includes animated tutorials, can be opened from within Designer using Help > Contents on the main toolbar.

Support

As with all successful control products, support is crucial and the team at ETC will do everything possible to ensure that your project is a success. Please do not hesitate to contact us with your questions, bug reports and suggestions at:

T: (800) 688-4116
E: service@etcconnect.com

Please also visit our website to keep up to date with the latest product news and software releases: www.etcconnect.com.

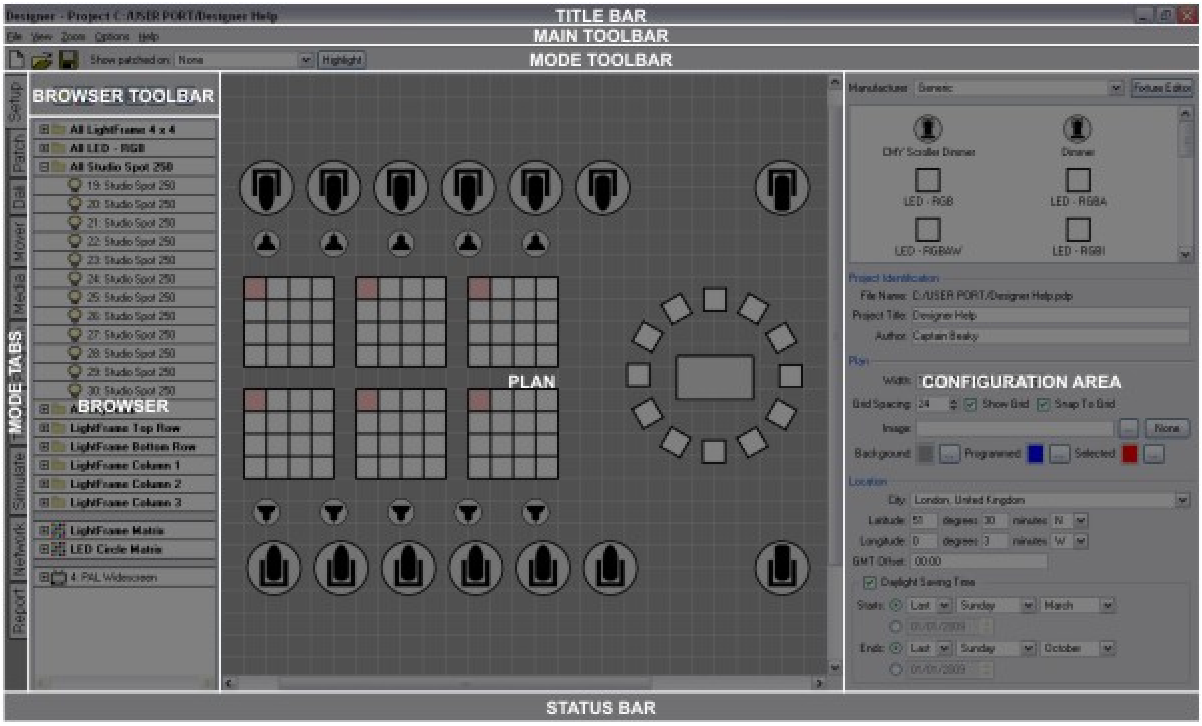
What's new in v1.11

Controllers

- Support for TPC-RIO

User Interface

The software has been designed to present a consistent graphical user interface and so it is worth familiarising yourself with the layout of a typical window before proceeding further:



Manufacturer: Generic

Project Identification

File Name: C:/USER PORT/Designer Help.pdp

Project Title: Designer Help

Author: Captain Beaky

Plan

Width:

Grid Spacing: 24 ☑ Show Grid ☑ Snap To Grid

Image:

Background: Programmed: 🟦 Selected: 🟥

Location

City: London, United Kingdom

Latitude: 51 degrees 30 minutes N

Longitude: 0 degrees 3 minutes W

GMT Offset: 00:00

☑ Daylight Saving Time

Starts: ● Last Sunday March

○ 01/01/2009

Ends: ● Last Sunday October

○ 01/01/2009

Main toolbar

The main toolbar contains the File menu to create New projects as well as Open and Save existing projects. The View menu accesses various optional,typically diagnostic, windows and the Zoom menu allows you quickly to manipulate the plan or timeline scale and position. The Options menu allows you to configure various application-wide options and the Help menu gains access to this help, software build details and the means to check for any software upgrades.

Mode tabs

The application is divided into ten modes which can be selected by clicking on the appropriate tab. See the Quick Start overview for a brief description of each mode and then the Reference section for more details.

Mode toolbar

The left hand end of this toolbar always carries the New, Open and Save project buttons. The remainder is populated with tools as appropriate to the selected mode.

Browser

The browser is common to most modes and provides the primary interface for selecting, expanding and grouping fixtures in the project. The rows of the browser provide the seeds for Designer's timeline programming interface. Some modes (Simulate, Network & Reports) have no browser since fixture selection is not relevant. Scroll bars will appear as required and the browser can be made wider by dragging the right hand border.

Browser toolbar

The Browser toolbar provides controls for expanding and collapsing groups and compound fixtures as well as for creating groups and Pixel Matrices.

Plan

A graphical representation of the installation that provides an interface for selecting fixtures and simulating the results of your programming. This plan simulation is one of Designer's most powerful features as it allows you to program and visually test your programming when off site. Indeed, this feature can be useful in presenting your ideas when bidding for a contract.

Configuration area

Depending on the mode and items selected, context-sensitive configuration or control pane(s) will appear here for fast and convenient editing.

Status bar

Context-sensitive status and progress is displayed in this area.

Keyboard shortcuts

For ease and speed of use various keyboard keys map to application commands, particularly with regards window navigation:

● Function keys	Press a function key (F1 thru F10) to change the application’s mode, akin to selecting a mode tab on the left hand side.
● Ctrl + F8	Press to open or close the Tear-off Simulator.
● Delete / Backspace	Press Delete or Backspace to delete the selected fixtures or groups of fixtures, a confirm dialog will appear if this will cause fixtures to be totally removed from the project with subsequent loss of programming. Press Delete or Backspace to delete the selected presets from a timeline.
● Esc	Press Esc to clear the fixture selection. Press Esc to abort a fixture move or timeline drag operation (preset move/resize).
● Cursor keys	Use the cursor keys to nudge a fixture selection (Setup) or scroll a matrix or timeline (Media, Program).
● Shift + cursor key	Use to super-nudge a fixture selection (Setup).
● Page Up / Down	Use Page Up and Page Down to zoom in and out of a plan, matrix or timeline.
● Tab	Use to select the next fixture (with a single fixture selected in Setup, DALI & Mover) or next editable cell (on a properties tab for example).
● Shift + tab	Use to select the previous fixture (with a single fixture selected in Setup, DALI & Mover).
● Ctrl + tab	Use to select the next timeline for editing (Program).
● Shift + Ctrl + tab	Use to select the previous timeline for editing (Program).
● Ctrl + F	Use to fit the plan or timeline to the screen.
● Ctrl + T	Use to display the plan at actual size (1:1 pixel mapping).
● Ctrl + click	Hold Ctrl while clicking to select multiple fixtures (Setup, DALI & Mover) or presets (Program).
● Shift + click	Hold Shift while clicking to select ranges of fixtures (Setup, DALI & Mover).
● Ctrl + drag	Hold Ctrl while dragging a preset on a timeline to snap to the start/end of other placed presets (Program) or while dragging a fixture selection to copy the selection using the same relative layout (Setup).
● Shift + drag	Hold Shift while dragging a preset on a timeline for finer resolution (Program) or while dragging a fixture selection to invert the Snap To Grid behaviour (Setup).
● Ctrl + A	Select all fixtures (Setup, DALI & Mover) or presets (Program).
● Ctrl + N	New project.

● Ctrl + O	Open project.
● Ctrl + S	Save project.
● Ctrl + E	Export project (exports the project including media & plan background image for transfer or backup purposes).
● Ctrl + U	Upload to the Controller(s).
● Ctrl + shift + U	Save Snapshots.
● Ctrl + Q	Quit the Designer application.
● Spacebar	Start or pause the simulator.
● F	Drops a flag in learn timing mode

Notes for Mac OS X users

Unison Mosaic Designer makes a good deal of use of the two button mouse with right-click being used to invoke context-sensitive dialogs. As the majority of Mac users have only a single button mouse they must hold Ctrl while clicking to get this functionality. In most cases Macs use the Apple key instead of the Ctrl key except for moving between timelines in the Program tab where Ctrl is used due to an OS Xkeyboard shortcut change. Shift and Alt work as described for Windows.

Overview

The Designer software is the tool provided to configure and program the Unison Mosaic Controllers and Remote Devices. The Controllers have been designed specifically for the architectural and installation markets and, as opposed to DMX frame store solutions, offer genuine lighting, audiovisual and show control functionality.

Lighting & video is programmed on timelines, with a particular timeline having control data for one, some or all the lighting fixtures being controlled. Multiple timelines are supported and so a single unit can control multiple distinct zones, or more complex presentations can be programmed with external triggers coming from multiple systems.

The software offers powerful functionality with a simple and intuitive graphical user interface. Most operations can be performed with mouse clicks (typically left-click for selection and right-click for context sensitive options & commands) and drag-and-drop. Creating a project is broken down into ten sections, use the mode tabs down the left hand side of the application or the function keys (in brackets) to toggle between them:

Setup (F1)

In Setup you add your fixtures to the plan, arrange them in groups or customize their behaviour. Use the Properties pane to import a bitmap for the plan, set the plan's size and grid spacing and specify the geographic location of the installation. See Setup reference.

Patch (F2)

In Patch the fixtures are assigned to the connected Controllers (see Network) and given DMX universes and addresses. This step can be skipped during design and only completed during installation. See Patch reference.

DALI (F3)

In DALI you patch and define DALI groups & scenes for any DALI ballasts in the project. Unlike DMX fixtures, these definitions are stored in the DALI ballasts themselves and so the configuration must be uploaded separately from here. See DALI reference.

Mover (F4)

In Mover you program presets for any automated lights with position or beam manipulation controls. These presets can then be placed onto timelines along with other programming. See Mover reference.

Media (F5)

Media allows you to create virtual video screens and map fixtures to pixels of the screen. Here you also import and manage the media files (either static images or video) which can then be played back on these screens and any audio visual fixtures in the project. See Media reference.

Program (F6)

Program is where you create and edit the timelines that make up your presentation. Each fixture or group of fixtures is a row of the timeline and you can drag-and-drop from an extensive range of built-in intensity and colour effects, as well as placing mover presets and media clips. See Program reference.

Trigger (F7)

In Triggers you connect your programming with the real world. At its most basic you can define which timeline to begin on startup but for more complex environments with external triggers you can define a detailed script, even incorporating conditions if necessary. See Trigger reference.

Simulate (F8)

Simulate allows you to view a representation of your project in plan format. You can play individual timelines to check your programming then run the whole project including triggers. A set of buttons allow you to simulate external triggers in order to test your programming properly. See Simulate reference.

The Simulator can also be opened in its own window so that it is permanently available, typically on a second monitor. Tear off (click and then drag to the right) the Simulate mode tab or press Ctrl (Apple) + F8 to open this window.

Network (F9)

This is where you manage your Unison Mosaic hardware, assigning connected Controllers to the Controllers in your project, configuring their input/output interfaces and any attached Expansion Modules or connected Remote Devices. See Network reference.

Report (F10)

Here you can view and organise spreadsheet-style reports listing the elements within the project for example fixture schedule, patch, triggers. This is useful in providing documentation about the installation for future reference and to aid maintenance. Reports can be exported as *.tsv files (Tab Separated Values) for importing into an Excel spreadsheet for formatting and printing. See Report reference.

Quick Start

Creating a project

Having installed QuickTime and Designer, launch the application and select New Project from the Launch Project dialog, choose a location for the file to be stored and give it a memorable name. A blank project will then be created and the application will enter the Setup window ready for you to configure your plan, place your fixtures and start programming.

To create a simple Unison Mosaic Designer project we will consider these four stages:

1. Setting up your plan, fixtures and patch
2. Creating Media and Mover presets ready to use when programming
3. Programming your timelines
4. Creating the triggers that will operate your show

Getting started: Adding Conventionals, LEDs and Moving Lights to the plan and patching them

1. From the Fixture Library pane, select Conventional from the Generic manufacturer and drag to the plan.
2. Right click (Ctrl Click for Mac users) on the inserted fixture, click on Duplicate Fixture, edit Width to 6 and click Ok.
3. Return to the Generic Library, select LED-RGB and drag to the plan. Duplicate an 8x8 array.
4. From the library, use the drop down menu to select Robe, ColorSpot 1200, drag to plan and duplicate fixture again, this time 6 in a circle.
5. Notice the fixture browser on the left has populated each fixture in groups based on fixture type. The middle three tree buttons above the browser allow you to expand and collapse the items. Collapse all for now.
6. Next we will generate Pixel Matrices for applying 2-dimensional colour effects and media presets.
7. Drag a box around the LED array to select all fixtures. Click on the New Pixel Matrix button above the browser to create a new matrix. Rename it "LED Matrix".
8. Repeat with the moving lights to create another matrix and name it all "ML Matrix".
9. Click on the Patch tab on the far left hand side of the screen to change to the Patch window.
10. Drag each group on to the blank universe to patch all fixtures. (This step is not essential to program and simulate).

Creating Media & Mover presets

1. Click on the Mover tab at the left to change to the Mover view.
2. In the Mover Preset window click on Create New, and select the moving lights by dragging a box around them on the plan, or click on their group in the browser.
3. In the parameter window click on the Position, Colour and Beam tabs and choose values for these attributes.
4. To create a further presets, simply leave your current programming and click Create New.
5. Click on the Media tab at the left to change to the Media view.
6. Select your LED matrix. You can move the pixels in relation to each other and resize the array window if desired.
7. To import media click on Create New in the Media Preset window. The Designer installer provides sample media which is located in the Designer installation directory: \resources\media_samples.

Programming Timelines

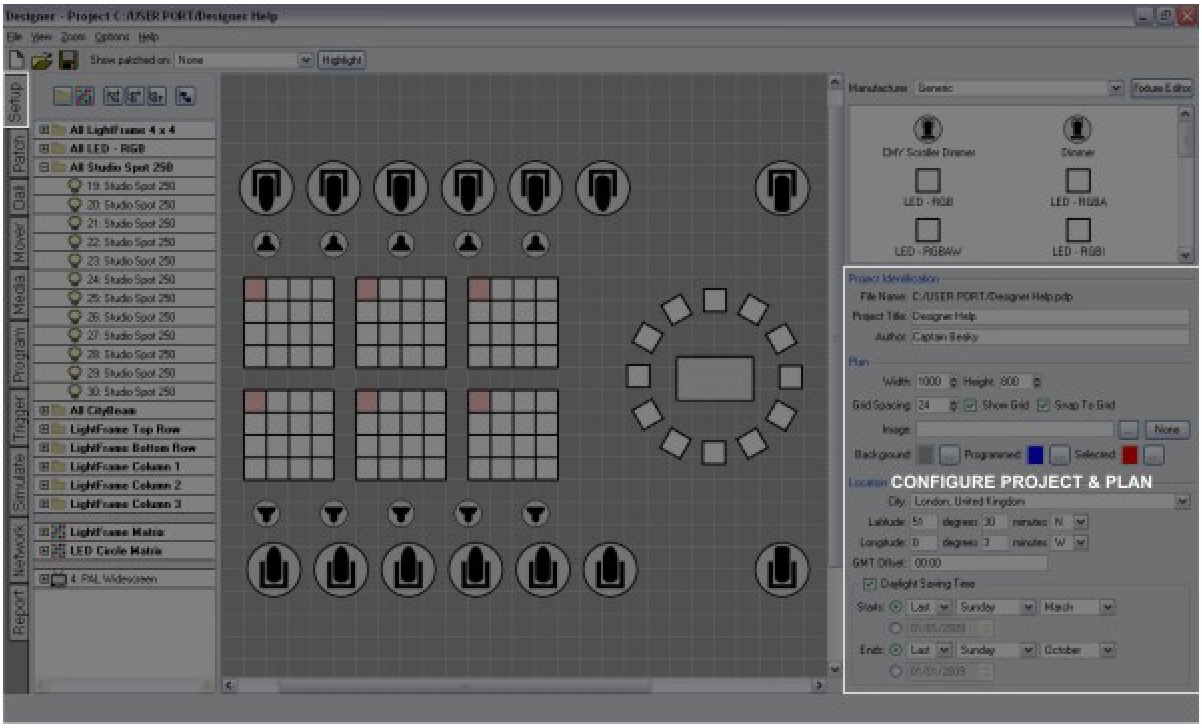
1. Click on the Program tab at the left to change to the Program view.
2. Note the fixture browser now has timeline rows associated with each fixture, group, or matrix, and two Mover rows have been created. Note also the preset directories on the lower right-hand side of the window: the Group directory includes built-in presets that can be applied to Group and Fixture rows, the Matrix directory includes built-in presets that can be applied to Pixel Matrices, the Media directory has been populated with any media you imported, and the Mover directory has been populated with the Mover presets you created.
3. Click on the Group directory tab. Select and drag the Fixed Colour preset to your All LED group, and drop at the start of the row. You can adjust the position and length of the effect on the row by dragging it on the timeline and selecting the handles at the end of the preset. The preset properties (in this case just colour) can be edited in the Properties pane. Fade times and characteristics can be edited in the Transitions pane.
4. Choose other presets from the Group directory, such as Rainbow and Sparkle, experiment with dropping them on your groups, or open up the group and apply to individual fixtures and edit parameters.
5. To view your timeline, drag the Simulate tab at the left onto the Program view to open the Simulate window on top of the Program window. Click on the play button to run the timeline.
6. Click on the Matrix directory. Drag these presets onto your LED Matrix row. Experiment with parameters and simulation.
7. Click on the Media directory. Drag your media clip onto your one of your Matrix rows. Experiment with parameters and simulation. (To experiment with the matrix size, e.g. to modify the resolution or the viewable area of the media, return to the Media window and adjust the dimensions of the matrix, the position of the pixels, etc.).
8. Click on the Mover directory. Drag your Mover presets to one of the Mover rows. Click on the Group directory and drag intensity to the fixture / group row to apply intensity. For CMY fixtures, apply colour effects.
9. Use the intensity preset to program the Conventional fixtures.
10. Create additional timelines by clicking on New Timeline. Name as appropriate and program.

Creating Triggers

1. Click on the Trigger tab at the left to change to the Trigger view.
2. Note that Trigger, Condition and Action directories are to the left of the worksheet area, to the right are properties panes for editing parameters, selecting components, etc.
3. To create a Startup trigger, click Start up in the Trigger directory and drag into the main worksheet area. Give it a descriptive comment. Drag Start Timeline from the Actions directory to the new Startup trigger row on the worksheet, and under Action Data in the Actions pane select a timeline from the dropdown menu.
4. Repeat, experimenting with different Triggers and different Actions. Note Change button for Real Time parameters, and Edit for the MIDI message builder. Question marks (???) on the worksheet will indicate where parameters or values need to be set
5. To create a conditional trigger, create a Digital Input trigger. Set it to input 1, drag a Start Timeline Action and select a timeline to play. Copy and paste the trigger (right click / Mac Ctrl+click). On the earlier trigger drag an Astronomical condition which will default to After Sunset. Set Action to play a different timeline. This achieves a condition option and a catch-all option if the condition does not apply.
6. Return to Simulate. Select Project above the control area. All the triggers you have created appear below. Clicking Play will activate the Startup trigger. Run your project, utilizing the triggers to mimic contact closures, real time triggers, serial messages, etc.

Setup - Project properties

With no fixtures selected, the Project Properties pane is displayed:



Project Identification

File Name: C:/USER PORT/Designer Help.pdp
Project Title: Designer Help
Author: Captain Beaky

Plan

Width: 1000 Height: 800
Grid Spacing: 24 ☑ Show Grid ☑ Snap To Grid
Image: … None
Background: ▨ … Programmed: 🟦 … Selected: 🟥 …

Location

City: London, United Kingdom
Latitude: 51 degrees: 30 minutes: N
Longitude: 0 degrees: 3 minutes: W
GMT Offset: 00:00
☑ Daylight Saving Time
Starts: ● Last Sunday March
○ 01/01/2009
Ends: ● Last Sunday October
○ 01/01/2009

CONFIGURE PROJECT & PLAN

Project Identification

The project filename (*.mdp) and path is displayed for reference.

Underneath are two fields for optionally entering a project title and the project's author, these fields are displayed on the Controller's web interface home page and are useful for reference once the installation completed.

If the title field is left blank the web interface will instead display the project's filename which may be useful for tracking iterative versions.

Plan

Size

The plan size can be set via the Width and Height fields (in pixels) and a solid background colour selected by clicking the Background browse button. The maximum plan size is 8192x8192 pixels.

Grid

The spacing of the working grid can be specified (again in pixels) and there are options to show this grid and whether fixtures should snap to it. This grid is useful in accurately and easily placing fixtures.

Background image

To use a background image click on the button next to the Image entry to browse for an image, either a Windows (*.bmp), Portable Network Graphics (*.png) or JPEG (*.jpg) image can be imported. It is envisaged that this image be a graphical representation of the installation, perhaps derived from architectural CAD drawings.

Use the Windows Alt + Print Screen command to take a screen shot of your CAD application and then use a bitmap editor to crop and resize the image to suit (see scale). Again, the maximum plan size is 8192x8192 pixels so make sure the bitmap is smaller or equal to these values. The plan size will automatically adjust to be that of this image.

When planning a fixture layout, give consideration to fixture selection and visualization. Particularly with large layouts, an abstracted arrangement of fixtures may be easier to view and work with than a pixel perfect, scale accurate representation.

Scale

The Unison Mosaic Designer fixture library uses a scale of 1cm:1pixel (0.394":1pixel) for the fixture icons so, for best results, the plan bitmap should be sized to this scale. If your installation is too large to be accommodated at this scale (i.e. bigger than 81.92m in either axis) then change the scale and use the Fixture Position settings to adjust the scale of your fixture icons accordingly.

Colour scheme

It may be desirable to change the colour of programmed and selected fixtures to aid clarity depending on the plan background colour or image, use the browse buttons to select appropriate colours. A darker background makes visualization clearer.

Location

City or Latitude/Longitude

At the bottom are the fields to set the location of the installation to ensure correct operation of the Controller's internal astronomical clocks. A city picker is provided to facilitate the coordinate entry but values can be entered directly into the Latitude and Longitude boxes - a web map service such as www.multimap.com is a useful resource for collecting this information.

Time zone

The local time zone can be entered as an offset to GMT, for example New York would be -05:00 being 5 hours behind GMT. If the city picker is used to select the location then the time zone will automatically be set.

NOTE: If such an offset is set then the Controller's date and time must be set to GMT not local time or this offset will be doubled.

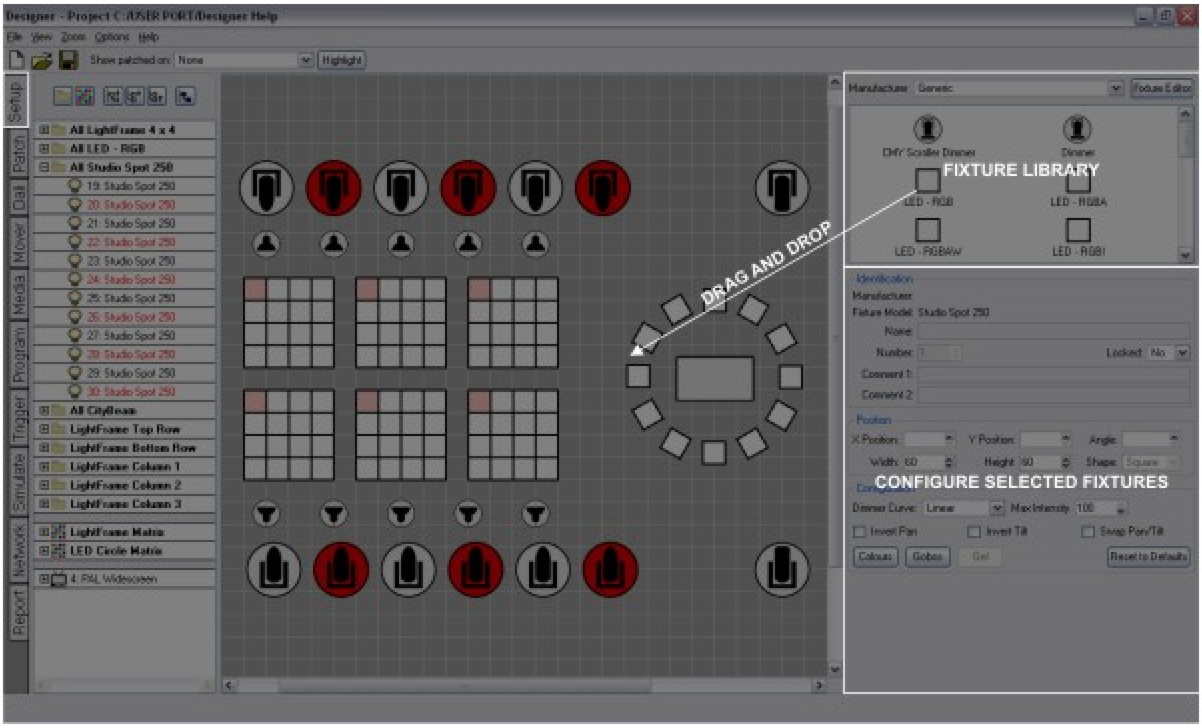
Daylight Saving Time

Check the Daylight Saving Time box to enable automatic DST adjustment. The rules for Daylight Saving differ by region but, if the city picker is used to select the location, the correct settings for that region should appear in the

Starts and Ends fields although it is recommended that you check that they are indeed valid. Alternatively, specific dates can be entered (the year is ignored).

Setup - Adding & organising fixtures

Once you have the plan setup as desired you can start populating it with the fixtures as required for the installation:



Manufacturer: Generic

Manufacturer:
Fixture Model: Studio Spot 250
Name:
Number: 1 Locked: No
Comment 1:
Comment 2:

X Position: Y Position: Angle:
Width: 60 Height: 60 Shape: Square

Dimmer Curve: Linear Max Intensity: 100
☐ Invert Pan ☐ Invert Tilt ☐ Swap Pan/Tilt

Fixture Library

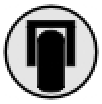
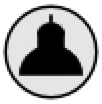
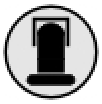
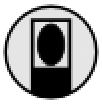
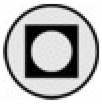
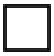
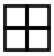
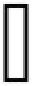
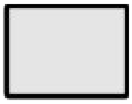
Unison Mosaic Designer incorporates a comprehensive fixture library grouped by manufacturer. A generic manufacturer is provided for standard fixtures such as dimmers, basic RGB LEDs and non-dim items that need to be switched such as fans or smoke machines.

A custom manufacturer is created when you make your own fixtures, either by right-clicking on a fixture within the library picker (not once placed on the plan) and selecting Customise Fixture to use this fixture as a starting point, or by pressing the Fixture Editor button at the top to create a fixture from scratch. Both operations will open the Fixture Editor.

Note however that common settings such as colour & gobo slots, size, shape and dimmer curve etc. can all be set on an individual fixture basis using the Fixture Configuration pane once the fixture has been deployed and selected, so creating a custom fixture library entry,a relatively complex process, may not be necessary.

Fixture icons & scale

The following icons are used to differentiate between fixture classes:

	Moving light - wash		DALI ballast (see DALI)
	Moving light - spot		Conventional fixture
	Moving light - mirror		Non-dim (switched control channel) or controller
	Accessory (eg. scroller)		Media server
	Discrete LED fixture (to scale)		Compound LED fixture (to scale)
	Fountain jet fixture		
	4:3 PAL/NTSC AV device		16:9 PAL/NTSC AV device

The LED and compound LED fixture icons are drawn to scale (1cm:1pixel) so that, coupled with a correctly scaled background image, the resulting plan and simulation is as realistic as possible. The other icons are drawn to a standard size that, in most cases, will produce a realistic result. All placed fixture icons can however have their size (scale) and even shape modified using the Fixture Configuration pane.

When using the Simulator these icons instead render the fixture's output, even displaying the selected gobo and iris settings for moving lights. Fountain jets simulate differently however, extending in length to mimic the jet of water.

Populating the plan

Simply chose a manufacturer, select the required fixture by clicking on it and then drag it onto its position on the plan, it will automatically be added to the Browser and grouped with all other fixtures of that type. Once placed, left click to select it, a red highlight will indicate the current selection, see selecting fixtures. Right click to delete, group or duplicate fixtures.

To add a fixture:

1. Use the drop down menu at the top of the library browser to select the manufacturer
2. Locate the required fixture
3. Click and drag the fixture onto the plan and release the mouse button to drop it (it will automatically be added to the Browser)

To duplicate a fixture (create an array):

1. Right-click on the fixture (on the plan not the Browser) to be duplicated
2. Select "Duplicate"
3. Select either "Rectangle" or "Circle"

4. Set the duplication parameters, see below
5. Press Ok

For rectangular arrays, positive width and height values will place the copies to the right and below respectively, negative to the left and above. Select either Rows or Columns to set the direction of the fixture numbering.

For circular arrays, select the radius, direction and count (number of fixtures) - complete circles are created in this way so, if arcs required, just delete those fixtures that are unwanted.

To copy a fixture or fixture selection:

1. Select the fixture(s)
2. Press and hold Ctrl (Apple)
3. Drag the copy to a new location on the plan and release the mouse button to drop (with multiple fixtures, their relative layout is preserved)

Note that pressing Ctrl *after* starting to drag will cause the selection to jump back to its original position and create a copy of the selection under the pointer.

To delete a fixture or fixture selection:

1. Select the fixture(s)
2. Press Delete or right-click > Delete
3. Press Delete to confirm (or Cancel to abort)

Note that the fixture(s) will be completely removed from the project and all programming discarded.

To see where a fixture is patched:

1. Check 'Show patched on' on the toolbar.
2. Move the cursor over a fixture - the fixture's patch will be shown next to the cursor.
3. Pick a Controller from the drop down list on the toolbar to see all fixtures patched to it - fixtures patched to the Controller will be shown in blue.

To highlight a fixture:

1. Select one or more fixtures using the Browser or the plan
2. Press the Highlight button, the fixture(s) will come on to their highlight defaults (typically open white)
3. Press Highlight again to turn off or select other fixtures to highlight

NOTE: The appropriate Controller must be on the network and correctly associated to highlight fixtures. Fixtures can also be highlighted from the universe tab in Patch.

DALI fixtures

DALI fixtures/ballasts are dragged onto the plan in the same way as all other fixtures but they do not populate the Browser and no groups are automatically made since DALI fixtures are programmed and controlled via dedicated DALI Interfaces, see DALI.

Audio visual (AV) fixtures

Select "Generic Video" from the drop down of manufacturers. Choose the appropriate type of device (PAL or NTSC,16:9 or 4:3) and drag it onto the plan, just as you would for a lighting fixture.

However, unlike other fixtures, this will automatically add a project AVC to control that device, see Controller Association. Similarly duplicating or deleting these AV fixtures will automatically add or delete AVCs as required.

It is important to appreciate that AV fixtures are treated differently to lighting fixtures in that they are directly coupled to an AVC (which takes the same name) and thus require no patching. The AV fixture and its AVC should be thought of as a single, integrated entity. See Working with the Audio Visual Controller for information about programming AV fixtures and AVC playback.

Import fixture plan

You can use File > Import Fixture Plan to import a fixture layout from a CAD application via a CSV file, see fixture plan file format.

Export fixture plan

You can use File > Export Fixture Plan to export a fixture layout to a CAD application via a CSV file, see fixture plan file format. If you have any fixtures selected you will be prompted to export all or only those selected.

Fixture Identification

With a fixture selected the top two fields detail the fixture's manufacturer (manufacturer id) and model (model id), they are for reference only and can not be edited.

Name & number

Here you can enter a new name for the fixture, useful to help make the browser easier to navigate, and the means to change the fixture's unique user number.

Every fixture added to the project is assigned a user number which is used as a shorthand method of selecting it, using the web interface's command line for example. Use the up and down arrows to change the number but note that only available numbers are shown so you may need to change the number of another fixture first to make that number available. Note that the user number does not affect the order of the fixtures in the Browser and thus the order used for transitions.

Locked

Select "Yes" to prevent the fixture(s) from being moved or included in drag selections. Evidently, once locked, drag selection is prohibited to select multiple fixtures to unlock so you must use the Browser instead.

Comments

Below this are two fields for entering any comments about the fixture, useful for annotating the project's documentation. These comments will appear in the fixture report and in exported CSV fixture plans.

Fixture Position

Use these fields to set numerically the fixture's position and orientation on the plan and to change the size and shape of the fixture's icon, useful for tweaking their scale to that of the plan background if the default scale not used (1cm:1pixel).

It is desirable to position the fixtures on the plan as accurately as possible to improve both the accuracy of the programming (in particular pixel matrices created automatically from the plan layout) and the general neatness of the project and simulation.

Grid

Use the plan properties to set the spacing of the plan's grid, whether it is displayed and whether fixtures should snap to it. Holding Shift while dragging a fixture selection will invert the snap to grid behaviour.

Nudge

Use the cursor keys to nudge a fixture selection up, down, left or right by the amount set as the grid spacing. Alternatively, use the up and down arrows by each of the fixture's position fields. Holding Shift while using the cursor keys performs a super-nudge of 10x the grid spacing.

Alignment

Fixtures can be aligned to one another by making the selection and entering the value to be shared into the appropriate position field.

Fixture Configuration

This pane allows you to configure the selected fixture(s):

Intensity

The fixture's dimmer curve and maximum intensity can be set, use the Dimmer Curve pull-down to determine the type of cross fade the intensity channel will perform and set a Maximum Intensity level, useful for balancing light output.

DALI

DALI ballasts can be configured as allowed for by the DALI standard; Min Level (0>254), Max Level (0>254), Power On Level (1>254) and Bus Failure level (No change, 0>254). The standard specifies a level range of 0>254 with 255 being used as a special case meaning "no change", a mask if you like. Unlike DMX fixtures, these settings are stored in the ballasts themselves and so must be uploaded separately, see DALI.

Ballasts can also have their default Fade Time and Fade Rate set in the configuration pane.

Emergency DALI ballasts will also have the option to the set the Prolong time in the configuration pane. More information about emergency ballasts can be in the DALI topic.

NOTE: The default Fade Time and Fade Rate will be overwritten when new values are sent to ballasts during playback from triggers or programming. This is due to the way DALI ballasts store this information.

Moving lights

Moving lights can be customised for the project as one would on any sophisticated moving light console. Use Invert Pan, Invert Tilt and Swap Pan & Tilt to normalise the way they respond to the position controls.

Customise the fixture’s gobo & colour wheels by pressing the Gobos or Colours buttons to open the Configuration dialogs. Drag from the library onto the correct slots as required, press Ok to save or Cancel to abort.

Gel colour

For those working with gelled lights it is possible to simulate the gel’s colour so that the fixtures are rendered correctly, press the Gel button and select the required colour via the colour picker.

Reset to Defaults

Use this to force the fixture to be redefined from its library definition, losing local changes and thus restoring it to its defaults. This is useful for updating fixtures on the plan with any library definition edits, forcing a redraw. Local changes to a fixture's geometry (shape, size) will be overwritten.

Highlight Patched Status

On the toolbar, use the Show patched on pull-down to select a Controller and then press Highlight to highlight the fixtures that are patched to that Controller. Select None instead of a Controller to highlight unpatched fixtures. Press Highlight again to turn it off.

Selecting fixtures

There are various ways of selecting fixtures which are common to all windows so it is worth covering them now:

Browser

The Browser is the most powerful and flexible method of selecting fixtures. Click on a group heading to select all fixtures within the group, expand a group by clicking on the plus sign and click on fixtures within to select individual fixtures and, with compound fixtures, expand them to select the individual pixels within. Fixtures and pixels are shown in red when selected.

Hold down Shift while clicking to select all contiguous groups/fixtures/pixels between clicks and hold down Ctrl (Apple on Mac) while clicking to select multiple non-contiguous individual groups/fixtures/pixels. Hold down Ctrl (Apple) while clicking to deselect a selected group/fixture/pixel.

Pressing Esc or clicking “in space” (anywhere on the Browser that isn’t a fixture) clears the selection.

The Browser also provides the interface to view and change the ordering of fixtures/pixels within groups. This order is used by the application to determine cue timing and effects skews, simply drag fixtures about within the Browser to change this order.

Plan

Only fixtures and pixels can be selected using the plan, to select groups you must use the Browser. Fixtures and pixels are shown in red when selected.

Shift and Ctrl (Apple) work with clicking as described above to select/deselect and you can also lasso fixtures by clicking and dragging around them, fixtures must be wholly enclosed to be selected.

Hold down Alt to select individual pixels within compound fixtures. Hold down Alt and Ctrl (Apple) to select/deselect multiple pixels.

Pressing Esc or clicking “in space” (anywhere on the Plan that isn’t a fixture) clears the selection.

Select next/previous fixture

With a single fixture selected, the Tab key will select the next fixture (next higher fixture number) and Shift + Tab will select the previous fixture (next lower fixture number).

Select all fixtures

Ctrl + A will select all fixtures.

Groups

Groups are an important concept to grasp as they serve three purposes:

Firstly, as you will see later, it is the rows of the Browser that make up the rows of the Program timeline interface thus it is convenient to gather fixtures/pixels that are to be programmed together into a group to simplify this procedure.

Secondly, as the order of fixtures/pixels within a group determines how programming and timing is rendered, it is sometimes useful to make multiple groups of the same fixtures with different ordering.

Finally, Groups can be used to set up intensity control zones in the Triggers window.

To create a group:

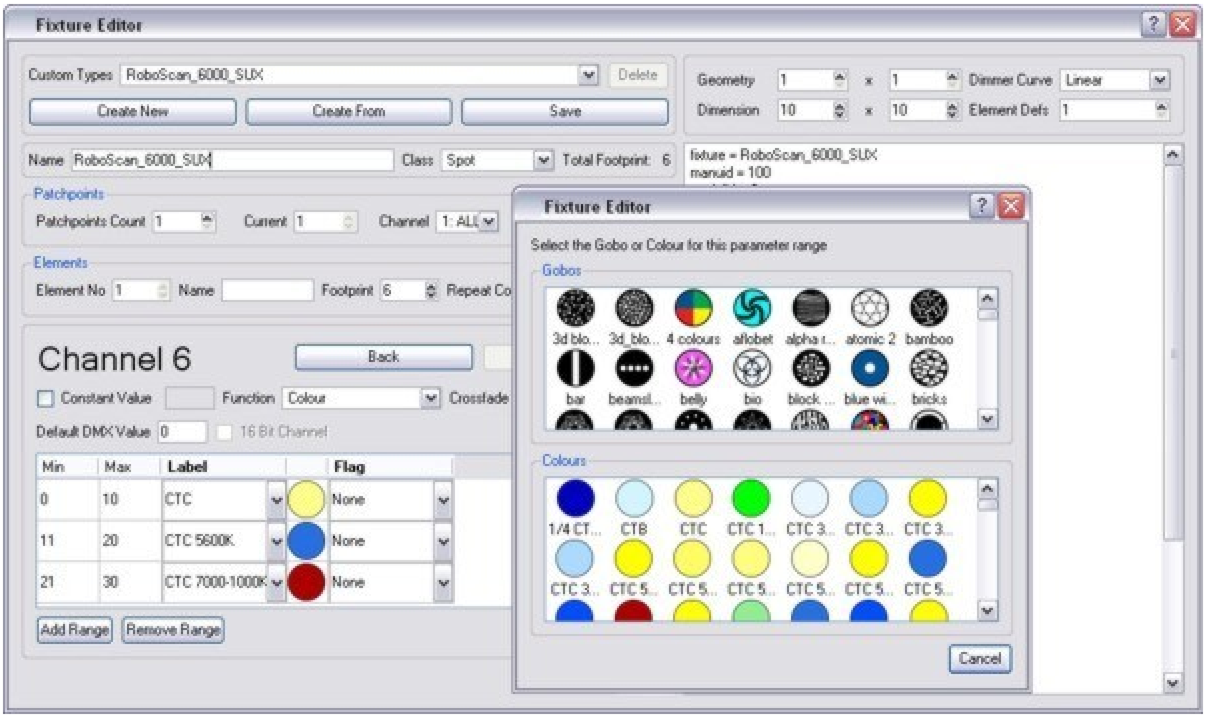
1. Select the fixtures you want to group using the Browser, the Plan or both - the order you do this in determines the order within the group
2. Right-click a member and choose Create Group From Selection or press the New Group button at the top of the Browser
3. Name the group which has been created in the Browser containing those fixtures

Alternatively:

1. Press New Group with no fixtures selected
2. Name this empty group
3. Drag fixture/pixel selections into the group from within the Browser - the order you do this in determines the order within the group

Setup - Fixture editor

While we endeavour to provide a comprehensive library of DMX compatible fixtures, a Fixture Editor is provided either to customise existing fixtures or create new ones:



Custom Types: RoboScan_6000_SUX

Geometry: 1 x 1 Dimmer Curve: Linear

Dimension: 10 x 10 Element Defs: 1

Name: RoboScan_6000_SUX Class: Spot Total Footprint: 6

fixture = RoboScan_6000_SUX
manuid = 100

Patchpoints Count: 1 Current: 1 Channel: 1: ALL

Element No: 1 Name: Footprint: 6 Repeat Co

Channel 6

☐ Constant Value Function: Colour Crossfade

Default DMX Value: 0 ☐ 16 Bit Channel

Min	Max	Label		Flag
0	10	CTC	●	None
11	20	CTC 5600K	●	None
21	30	CTC 7000-1000K	●	None

Select the Gobo or Colour for this parameter range

Customised or newly created fixtures will appear in the custom manufacturer folder. However, while we build up our database of fixture data we would actually prefer you to contact us to have your fixture added to the standard, distributed library. Please email your fixture request(s) to support.

IMPORTANT: When edits are made to the library definition of a fixture already in the project, the plan will not update until the fixture has been explicitly reset to its library defaults, see fixture configuration.

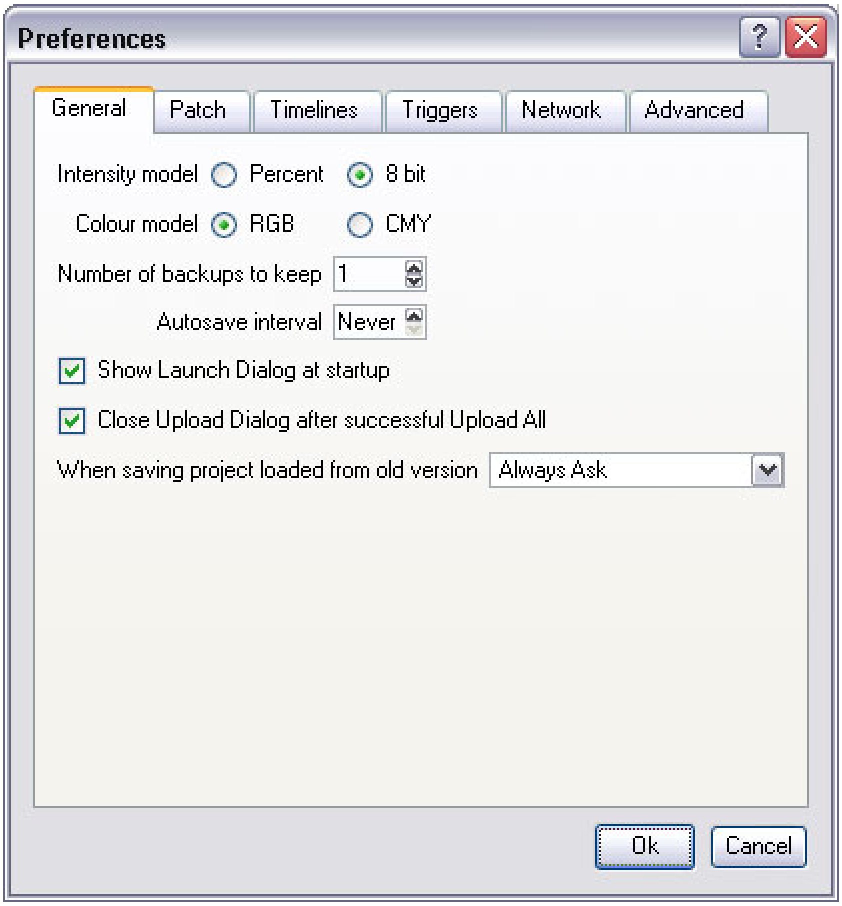
Electronic Theatre Controls, Inc. subscribes to the Carallon Automated Lighting Data Service.

Preferences

Select Options > Preferences on the main toolbar to open the Preferences dialog:

General

Select the General tab to change the default behaviour of Designer:



Intensity model ○ Percent ● 8 bit

Colour model ● RGB ○ CMY

Number of backups to keep 1

Autosave interval Never

☑ Show Launch Dialog at startup

☑ Close Upload Dialog after successful Upload All

When saving project loaded from old version Always Ask

Intensity model

Choose whether intensity and DALI presets are programmed using Percent (0>100%) or 8-bit (0>255) values.

Colour model

Choose whether colours are specified using RGB (additive) or CMY (subtractive) values, the latter might feel more natural for moving light console users.

Number of backups

Designer can keep a number old versions of the project file when you save and it is here that you set the number of old files to keep. Before saving your project (File > Save or Ctrl+S), Designer will rename the project file on disk by adding the current time and date to the file name, such as "my_project_bak_2007-04-18_15-58-09.mdp". If you already have the specified number of backups, the oldest backup will be removed from the disk.

Use File > Save As to produce manual backups of the project at each important programming milestone.

Autosave interval

Designer can periodically create an automatic backup of your programming and it is here that you turn this feature on and set the backup interval in minutes. Designer will maintain a backup file named "my_project_auto.mdp" which can be opened to retrieve recent programming in the event of a software crash. This file will be deleted if the Designer project is closed normally.

Show Launch dialog

Choose whether the Launch dialog is automatically opened at startup (default), uncheck this box to suppress this dialog.

Close Upload Dialog after successful Upload All

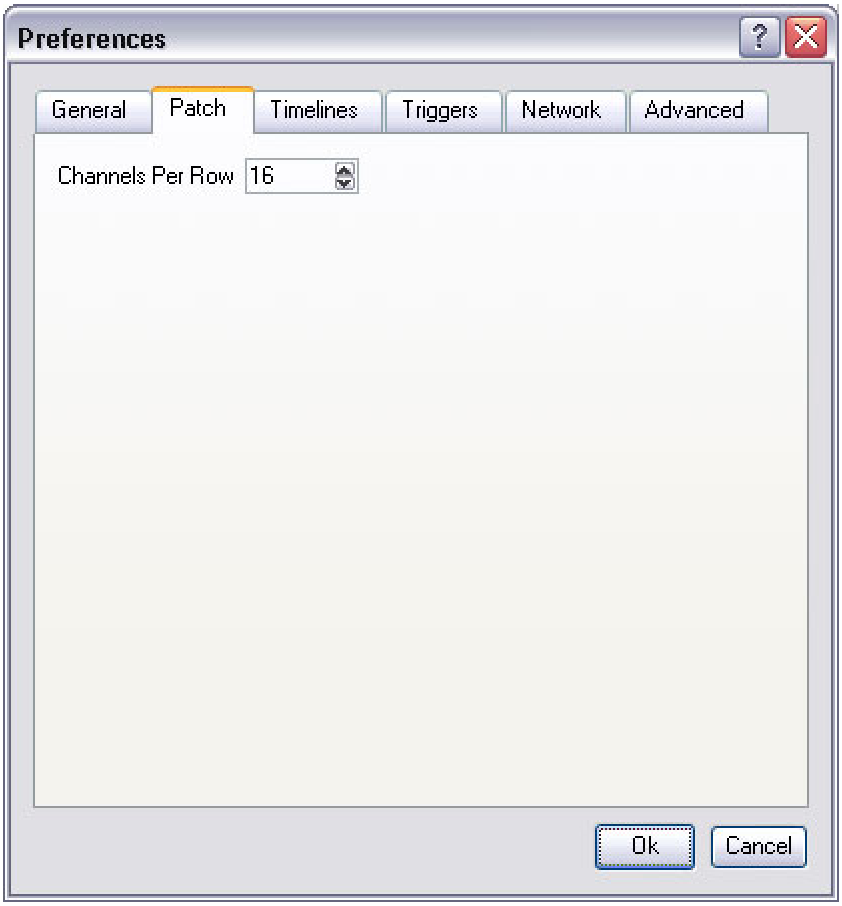
Check this to have the Upload Dialog close automatically after a successful Upload All.

Saving Projects from Old Version

Choose what Designer should do when saving projects that were last saved in an earlier version of the software. This can be a useful prompt to keep a backup of your work.

Patch

Select the Patch tab to change the default settings for patching:



Preferences

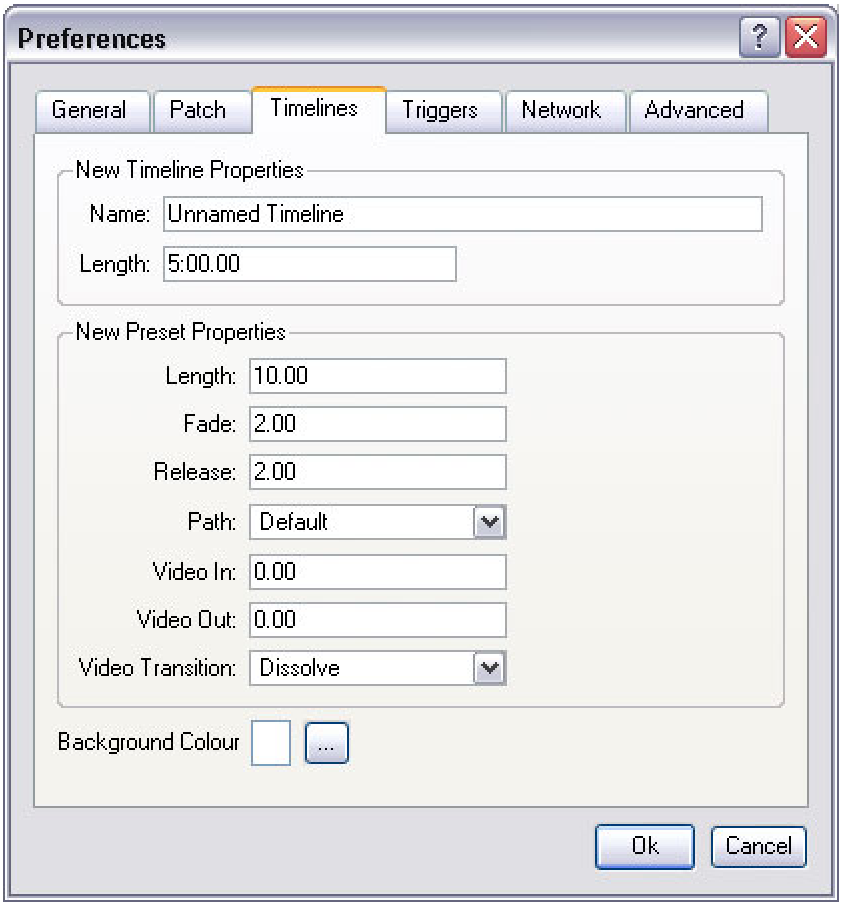
Channels Per Row 16

Channels per row

The Channels per row entry box lets you determine how many DMX channels per row are displayed which is useful for organising the display for complex fixtures; set this number to be a multiple of the number of channels a fixture uses to get a neater, tabulated display.

Timeline

Select the Timelines tab to change the default settings for timelines and presets:



New Timeline Properties

Name: Unnamed Timeline

Length: 5:00.00

New Preset Properties

Length: 10.00

Fade: 2.00

Release: 2.00

Path: Default

Video In: 0.00

Video Out: 0.00

Video Transition: Dissolve

Background Colour

New timeline properties

Specify the default name and length for subsequently created timelines, see timeline properties.

New preset properties

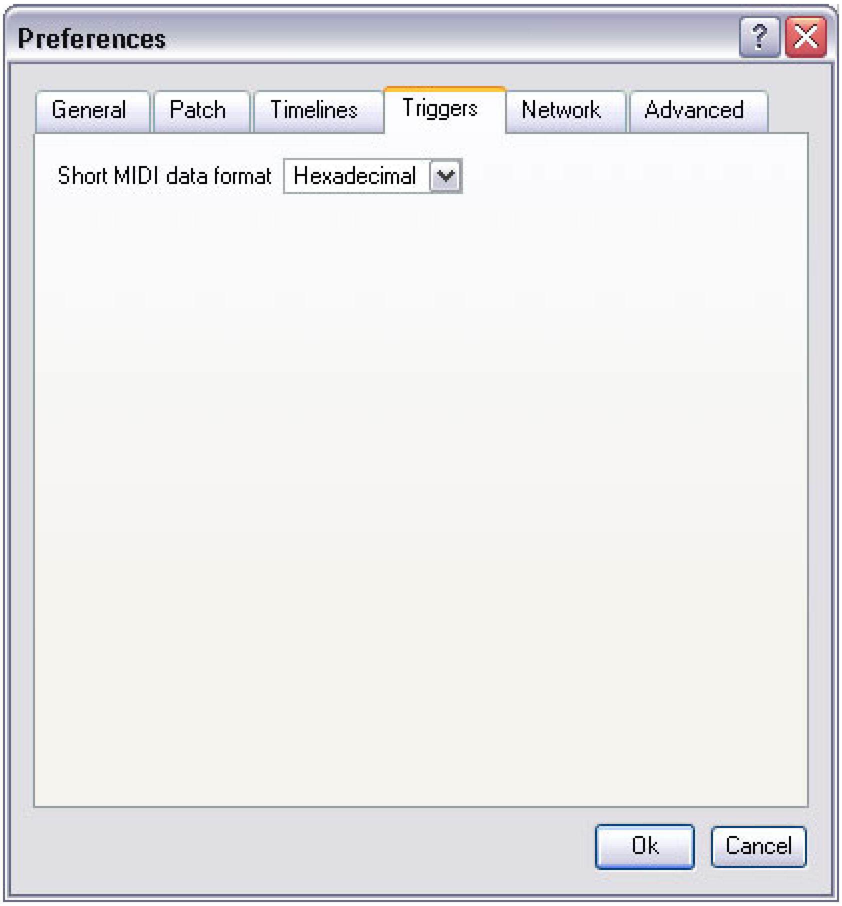
Specify the default properties for subsequently added presets, see preset types and properties.

Background Colour

The background colour of the timeline area of the Program window can be chosen here, press the button and select a colour. This is useful to make certain types of programming stand out better, for example a project using mainly intensity presets may be clearer with a dark grey background.

Triggers

Select the Triggers tab to change the default settings for triggers:

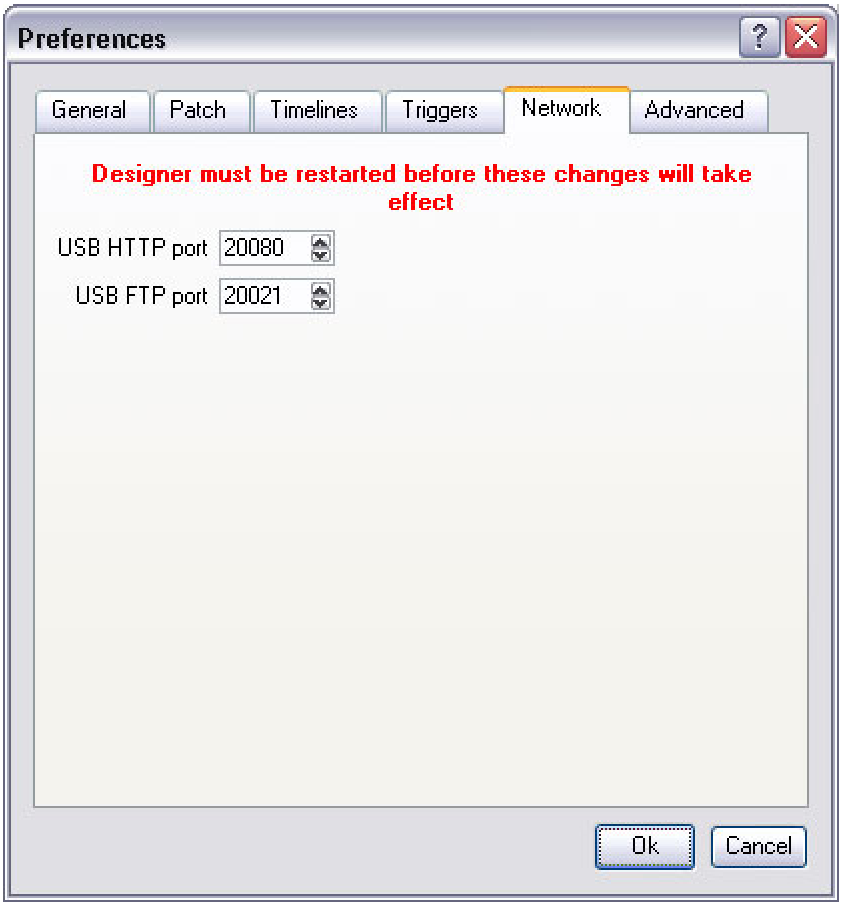


Short MIDI data format: Hexadecimal

Short MIDI data format

Select the default Short MIDI message data format to be either Decimal or Hexadecimal.

Network



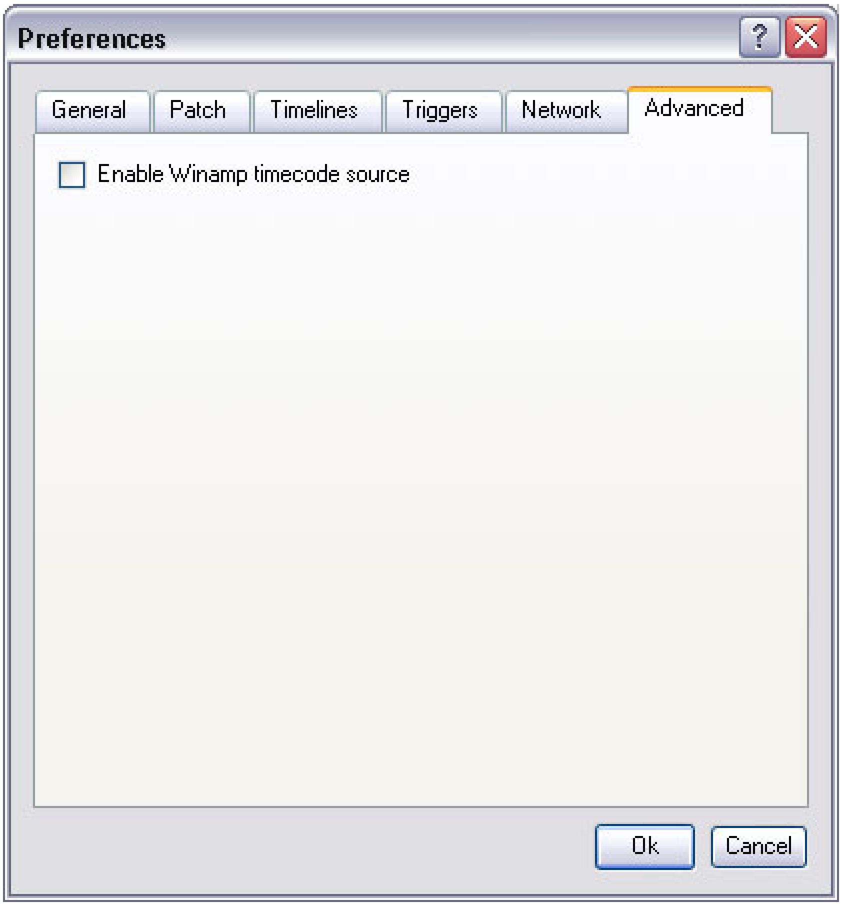
Designer must be restarted before these changes will take effect

USB HTTP port 20080

USB FTP port 20021

The web interface (HTTP) and file transfer (FTP) ports used when an MSC is connected via USB can be configured.

Advanced



□ Enable Winamp timecode source

Enable or disable using Winamp as a timecode source in Simulate. This feature is available on Windows only.

Patch

Once you have created your plan and added your fixtures you need to patch them, that is to say connect them to real fixtures via the appropriate Controller (MSC 1, 2, 4 or X), interface (port), protocol and address. Patching is optional for programming and simulation but fixtures must be patched eventually for the MSCs to control them, including using Output Live in the Simulator.

NOTE: The AVC does not require patching.

Before we cover patching in detail let's look at some of the terms used:

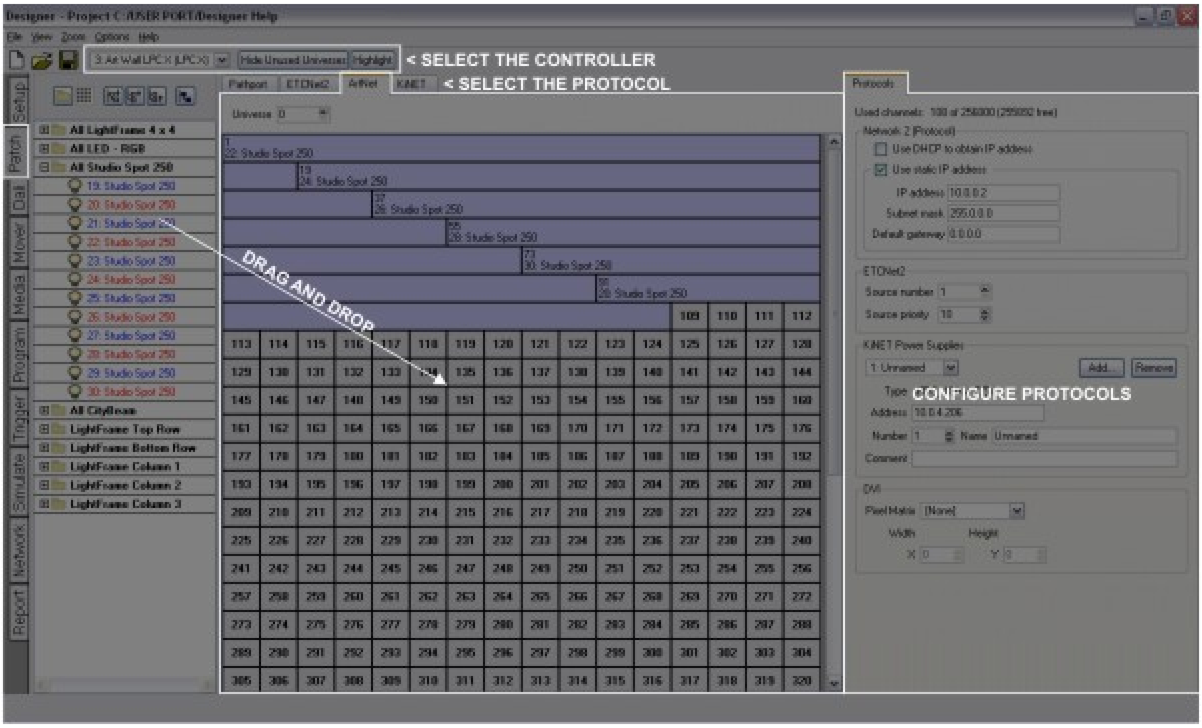
Patch terminology

Term:	Description:	For more information:
DMX	A digital serial control protocol for entertainment lighting. Officially called DMX512-A, it was developed by the USITT and has become the standard protocol for entertainment lighting control using the RS485 physical layer.	USITT DMX512
RDM	Remote Device Management, an extension of the USITT DMX512 protocol that supports bi-directional communication with dimmers & fixtures.	RDM
eDMX	A shorthand term for DMX-over-Ethernet protocols, see KiNet, Art-Net II, Pathport and sACN below.	
KiNet	A proprietary Ethernet control protocol developed by Color Kinetics (now Philips Solid State Lighting) used to control only their Ethernet PSUs.	
Art-Net II	A DMX-over-Ethernet protocol developed by Artistic Licence and widely used in the entertainment industry to distribute multiple universes of DMX data.	
Pathport	A DMX-over-Ethernet protocol developed by Pathway Communications and widely used in the entertainment industry to distribute multiple universes of DMX data.	
sACN	Streaming ACN (Advanced Control Network), a DMX-over-Ethernet protocol developed by ESTA to distribute multiple universes of DMX data.	ESTA
RIO	The ETC Remote Input Output 80/44/08 can output up to 96 channels of DMX per unit. See here for more information on patching to a RIO.	
DVI	Digital Video Interface, a standard for the delivery of digital video data to computer monitors. Used by certain LED manufacturers (for example Barco and Martin Professional) to drive their LED controller products.	
DALI	Digital Addressable Lighting Interface, a digital serial control protocol for architectural lighting. Developed by Philips Lighting it has become a standard: IEC 60929. DALI fixtures are not patched using this window, see DALI.	DALI
Universe	A common term given to a single DMX data link or port. A DMX universe carries 512 channels of control data each with 8 bit resolution. A single dimmer will use one channel while more complex fixtures will use multiple channels as required.	
Port	The KiNet equivalent to a universe.	
DMX Address	The term used to determine which of the 512 control channels of a DMX universe a fixture should look at to take its own control data. This "start address" must be set on the fixture or dimmer rack itself as well as patching the control system.	

Patch window

This window comprises three sections, to the left is the Browser, to the right the Protocol configuration pane, with the rest of the window being a graphical representation of a protocol's port or universe. The number of address columns displayed per row can be changed using Options > Preferences > Patch.

If you are using an MSC X or choosing to output eDMX from an MSC 1 or 2 then you must use the Protocols pane to configure these protocols, see Controller Protocols.



< SELECT THE CONTROLLER

< SELECT THE PROTOCOL

DRAG AND DROP

CONFIGURE PROTOCOLS

- All LightFrame 4 x 4
- All LED - RGB
- All Studio Spot 250
  - 19: Studio Spot 250
  - 20: Studio Spot 250
  - 21: Studio Spot 250
  - 22: Studio Spot 250
  - 23: Studio Spot 250
  - 24: Studio Spot 250
  - 25: Studio Spot 250
  - 26: Studio Spot 250
  - 27: Studio Spot 250
  - 28: Studio Spot 250
  - 29: Studio Spot 250
  - 30: Studio Spot 250
- All CityBeam
- LightFrame Top Row
- LightFrame Bottom Row
- LightFrame Column 1
- LightFrame Column 2
- LightFrame Column 3

Universe 0

1 22: Studio Spot 250
19 24: Studio Spot 250
37 26: Studio Spot 250
55 28: Studio Spot 250
73 30: Studio Spot 250
91 20: Studio Spot 250

Used channels: 100 of 256000 (255900 free)

Network 2 (Protocol)

□ Use DHCP to obtain IP address

☑ Use static IP address

IP address: 10.0.0.2

Subnet mask: 255.0.0.0

Default gateway: 0.0.0.0

ETCNet2

Source number: 1

Source priority: 10

KiNET Power Supplies

1: Unnamed — Add... Remove

Type

Address: 10.0.4.206

Number: 1 Name: Unnamed

Comment:

DVI

Pixel Matrix: (None)

Width X: 0 Height Y: 0

Patch toolbar & protocol tabs

Use the Controller pull-down to select the Controller for patching. Use the Hide Unused Universe button to suppress unpatched universes and the Highlight button to bring the selected fixture(s) to its highlight values, typically 100% intensity in open white for easy identification.

Use the tabs across the top of the graphical area to select the required protocol and then the universe or power supply & port pull-downs to select the target for patching which will be graphically displayed.

Patching DMX & eDMX fixtures

Simply select one or more fixtures in the Browser and drag them onto the required start address of the graphical representation. Right click on a patched fixture to unpatch it or clear the entire universe/port, drag it to move it (change its address). Fixtures may be patched to multiple addresses and universes/ports. Patched fixtures are shown in blue in the Browser, unpatched black.

To patch a fixture:

1. Use Controller and Universe to select the desired Controller and universe of this Controller (if the Controller has multiple universes)

2. Select the fixture in the Browser
3. Drag and drop the fixture onto the desired start address

To patch multiple fixtures:

1. Use Controller and Universe to select the desired universe
2. Select a group of fixtures in the Browser (see selecting fixtures)
3. Drag and drop the group of fixtures onto the desired start address of the first fixture in the selection

To change a fixture's address:

1. Use Controller and Universe to select the universe on which the fixture is patched
2. Select the fixture(s) on the universe layout and simply drag them to a new address - to move them to another universe you must unpatch (see below) then repatch

Fixtures may be patched to as many locations and universes as is required although typically a fixture will only be patched to one unique address. The Designer software will prompt you with a warning dialogue should you attempt to patch a fixture that is already patched, select Continue or Unpatch Existing as required. This prompt can be turned off if it proves aggravating.

However, you can not patch more than one fixture to the same address; a DMX channel can only be controlled by one parameter so fixtures can't overlap at all. If you drag one fixture onto another the incoming fixture will highlight in red to alert you that this address is already occupied. If you go ahead and drop it there anyway the software will prompt you whether to continue and unpatch the existing fixture(s) for you,select Ok to proceed or Cancel to abort. Again, this prompt can be turned off.

Some fixtures, for example the Vari*lite VL5, need to be patched twice since they have two distinct patch points, one for the intensity control (patched to the dimmer rack) and another for the fixture's automation controls.

To patch a multiple patch point fixture:

1. Locate the fixture in the Browser and expand it by pressing the plus sign to reveal its patch points
2. Use Controller and Universe to select the desired universe for the first patch point
3. Drag and drop the first patch point onto the desired start address
4. Repeat for the other patch point(s)

or

1. Patch the entire fixture as one
2. Drag the patch points apart to their desired addresses

To unpatch a fixture or multiple fixtures:

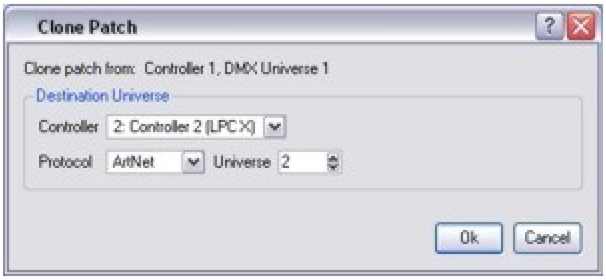
1. Select one or more fixtures using the Browser or the universe layout
2. Right-click and select Unpatch

To clear all patching from a universe:

1. Use Controller and Universe to select the universe to be cleared
2. Right-click anywhere on the universe layout and select Clear Universe

To clone a universe (copy the patch):

1. Use Controller and Universe to select the universe to be cloned
2. Press the Clone Universe button to open the Clone Patch dialog:



Clone Patch

Clone patch from: Controller 1, DMX Universe 1

Destination Universe

Controller: 2: Controller 2 (LPC X)

Protocol: ArtNet Universe: 2

3. Select the target Controller, Protocol and Universe using the pull-down
4. Press Ok to clone the patch to this universe or Cancel to abort

To highlight a fixture:

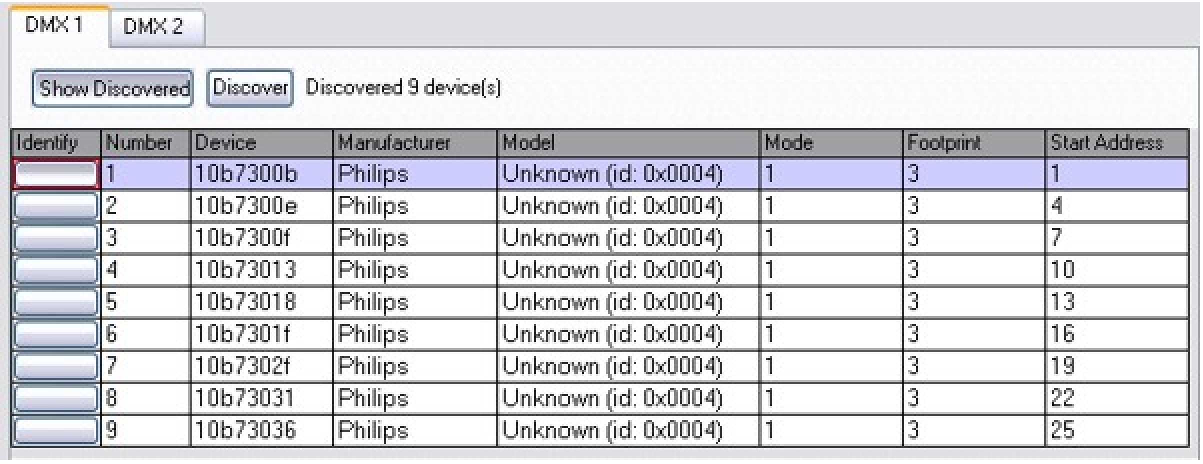
1. Select one or more fixtures using the Browser or the universe layout
2. Press the Highlight button, the fixture(s) will come on to their highlight defaults (typically open white)
3. Press Highlight again to turn off or select other fixtures to highlight

NOTE: The appropriate Controller must be on the network and correctly associated to highlight fixtures. Fixtures can also be highlighted from the plan in Setup.

To hide unused universes:

1. Select the Controller
2. Press the Hide Unused Universes button, only universes with patched fixtures will be displayed
3. Press Hide Unused Universes again to display all or select another Controller

RDM device discovery



Discovered 9 device(s)

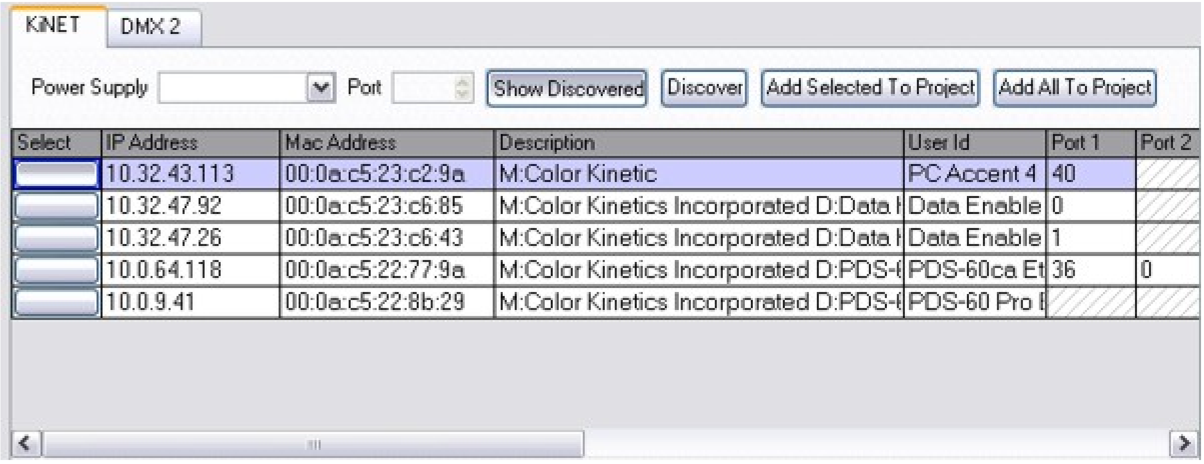
Identify	Number	Device	Manufacturer	Model	Mode	Footprint	Start Address
	1	10b7300b	Philips	Unknown (id: 0x0004)	1	3	1
	2	10b7300e	Philips	Unknown (id: 0x0004)	1	3	4
	3	10b7300f	Philips	Unknown (id: 0x0004)	1	3	7
	4	10b73013	Philips	Unknown (id: 0x0004)	1	3	10
	5	10b73018	Philips	Unknown (id: 0x0004)	1	3	13
	6	10b7301f	Philips	Unknown (id: 0x0004)	1	3	16
	7	10b7302f	Philips	Unknown (id: 0x0004)	1	3	19
	8	10b73031	Philips	Unknown (id: 0x0004)	1	3	22
	9	10b73036	Philips	Unknown (id: 0x0004)	1	3	25

In the DMX patch tab, press 'Discover' to find RDM-capable devices attached to the current Controller. The Discover button is only enabled if the current Controller is associated with a physical Controller and that Controller has been found on the network.

Toggle 'Show Discovered' to show and hide the discovered RDM devices.

To identify a device, toggle the button in the Identify column of the table. To readdress a device, enter a new value in the Start Address column.

KiNet device discovery



Select	IP Address	Mac Address	Description	User Id	Port 1	Port 2
	10.32.43.113	00:0a:c5:23:c2:9a	M:Color Kinetic	PC Accent 4	40	
	10.32.47.92	00:0a:c5:23:c6:85	M:Color Kinetics Incorporated D:Data I	Data Enable	0	
	10.32.47.26	00:0a:c5:23:c6:43	M:Color Kinetics Incorporated D:Data I	Data Enable	1	
	10.0.64.118	00:0a:c5:22:77:9a	M:Color Kinetics Incorporated D:PDS-(	PDS-60ca Et	36	0
	10.0.9.41	00:0a:c5:22:8b:29	M:Color Kinetics Incorporated D:PDS-(	PDS-60 Pro I		

In the KiNet patch tab, press 'Discover' to find KiNet power supplies.

NOTE: KiNet discovery is performed via the network interface of the computer running Designer, **not** via the Controllers. If the computer has more than one network interface then you will be prompted to select which network interface to use for KiNet discovery. The network interface must have an IP address in the 10.x.x.x range for discovery to be completely successful.

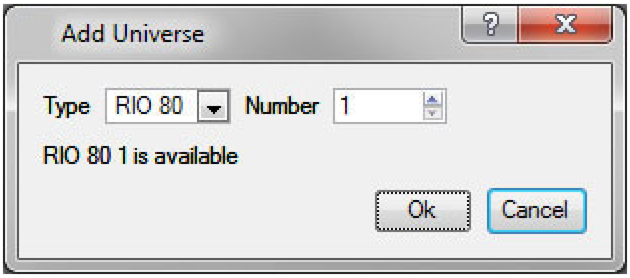
Toggle 'Show Discovered' to show and hide the discovered power supplies.

Power supplies that have an IP address that is already used by the current Controller will appear in grey.

To add a power supply to the current Controller, press the button in the 'Select' column and press 'Add To Project'. A new power supply will be added to the current Controller and that power supply will be automatically selected. If Designer doesn't recognise the type of the power supply, you will be prompted to select it from a list.

To add all the discovered power supplies to the current Controller, press 'Add All To Project'.

Patching fixtures to a RIO



Type RIO 80 Number 1

RIO 80 1 is available

In the RIO section of the patch tab you'll need to add a universe that will joined to a RIO before you can patch fixtures. Click 'Add Universe' and choose the RIO to which you want to output that universe of DMX.

The RIO is capable of outputting 96 channels of DMX. Any channels patched to a RIO will subtract from the maximum channel output capacity of the Controller. The RIO and Controller must be on the same network.

Patching DVI fixtures (MSC X)

To output data using the DVI port you must first create a pixel matrix that matches the LED controller's pixel map. Once this has been done, use the Pixel Matrix pull-down on the protocol tab to select which matrix will be output via the DVI port and select an X & Y offset as required. Any programming for the fixtures in the pixel matrix will now output on the DVI port, not just programming applied directly to the pixel matrix.

The MSC X's DVI port is set to a fixed 1024x768 @ 60Hz resolution which is compatible with most LED controllers. The LED controller (or monitor) MUST be connected when the MSC X boots or resets for the port to become active.

Used channels

At the top of the Protocols pane is an indicator of the number of used and available channels for the selected Controller.

When the Controller is an MSC X, and if you have not associated the controller with a physical device, then you are allowed to patch fixtures up to the capacity you have chosen (see Controller Association).

However, when you associate an MSC X with a physical device, you will only be able to associate to devices which have a capacity equal or greater than the patched fixtures. After associating with a device, you will be unable to exceed the capacity of that device.

eDMX Pass-Thru

When using an MSC in a project it is possible to allow eDMX from another eDMX source to be passed through to the controller’s DMX ports. With an MSC selected, the bottom of the protocols pane will show the eDMX Pass-Thru settings. Select which universe the DMX port will be transmitting. Note that with an MSC 2 you'll be able to choose a different universe for each DMX port on the controller. There is also the option to auto-revert to the project’s output if eDMX isn't received for a specified amount of time.

Note that only Art-Net and sACN are currently supported for eDMX pass-thru.

Art-Net output customisation

By default, a controller with less than 30 universes of Art-Net patched will broadcast all data until a device requests unicast for a specific universe. Controllers with more than 30 Art-Net universes patched will only unicast data to devices requesting universe data and will not automatically broadcast.

There is the option to 'Disable Broadcast' for a controller. The controller will still unicast data to devices that request it. There is also the option to 'Always Broadcast' on a per universe basis. This will force the controller to always broadcast that universe's data. 'Always Broadcast' will override the 'Disable Broadcast' option.

DALI

This window need only be used if there are DALI ballasts in the project and one or more RIO Ds or DALI-M Expansion Modules are being used.

Overview

The Digital Addressable Lighting Interface (DALI) is a digital serial control protocol for architectural lighting. Developed by Philips Lighting it has become a standard: IEC 60929. DALI differs from DMX in a number of important ways:

- Only 64 ballasts per DALI bus (interface)
- Only 16 groups per interface
- Only 16 scenes per interface
- Ballast configuration (including address), groups and scenes is uploaded to and stored in the ballasts themselves
- Topology-free DALI bus operates at very low data rates
- Command-based protocol, ballasts perform fades and maintain levels
- Only certain discrete fade values are permitted

As a result the DALI protocol is not suitable for rendering effects and media, programming is restricted purely to recalling lighting levels via the Set Level and DALI Scene presets, see DALI presets.

DALI interfaces

Each RIO D, TPC with TPC-RIO or DALI-M Expansion Module added to the system supports one DALI bus (up to 64 ballasts) and is assigned to a DALI interface within Designer. Due to the nature of the DALI protocol, these DALI interfaces are insular affairs with each having its own unique set of ballasts and groups. Each interface must be configured and uploaded to individually.

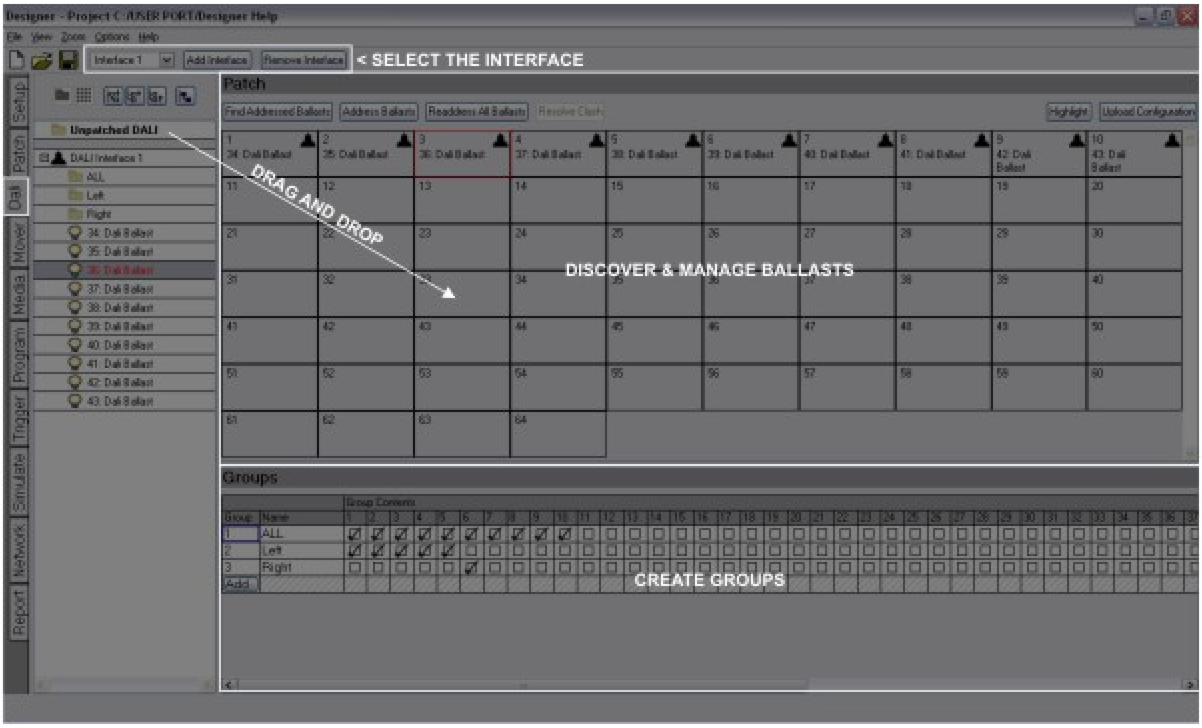
When you add a RIO D, TPC with TPC-RIO or DALI-M Expansion Module you select which DALI interface it should be assigned to. Only one device can be assigned to each interface. Use the Add Interface button on the DALI toolbar to add another. Use Remove Interface to remove an unwanted one.

Up to 100 DALI interfaces can be added to a project, but the following restrictions apply:

- A TPC or an MSC 1 can support up to 16 RIO Ds.
- An MSC 2 can support up to 32 RIO Ds.
- An MSC 4 can support up to 64 RIO Ds.
- An MSC X can support up to 100 RIO Ds.
- An AVC cannot support any RIO Ds.
- The original MSC 1/2 hardware each support up to two DALI-M Expansion Modules.

DALI addressing & grouping

Each DALI interface is configured separately, use the pull-down on the DALI toolbar to select an interface for configuration:



< SELECT THE INTERFACE

DRAG AND DROP

DISCOVER & MANAGE BALLASTS

CREATE GROUPS

The top pane of the DALI interface window displays the 64 real ballast address cells, the bottom pane the group membership matrix. The device commands are available to manage the ballasts on each DALI interface.

Device commands

IMPORTANT: Designer must be connected to the Controllers with RIO Ds or DALI-M Expansion Modules and the DALI ballasts must be active to perform these operations.

A DALI ballast internally stores its address, this is a number between 1 and 64. It is also possible that the ballast has never been addressed so it does not have an address. These commands are used to discover and address DALI ballasts:

Find addressed ballasts

This queries the ballasts attached to the device and reports all addressed ballasts found, an icon is added to each address cell when the corresponding ballast has been found.

Address ballasts

This finds all ballasts without an address and randomly assigns them a free address. It will not change the address of any already addressed ballasts.

Readdress all ballasts

This will clear the addresses of all ballasts and then assign every ballast a random address.

Resolve clash

It is possible that two ballasts can have the same short address. If that happens the ballasts clashing are shown with a red icon. The resolve clash button will move the clashing ballasts to a random address that is unused by any other ballast.

Identify emergency ballasts

Send all emergency ballasts a command to indicate their address on the multicoloured LEDs on the fixture. Whilst this is enabled the command will be sent every 10 seconds.

To manually readdress a DALI ballast:

1. Select the ballast icon in the current address cell
2. Drag it to the target (preferably empty) address cell

The ballast is readdressed to the target cell. If the target cell was occupied then the ballasts should swap addresses although this has not been found to be terribly reliable.

To highlight a ballast

Select an address cell (containing a ballast icon) and press Highlight to bring this ballast to full level, select another cell to highlight instead or press Highlight again to turn off.

Patching DALI fixtures

When you add DALI fixtures to your project plan in setup these "abstract" ballasts are assigned to the Unpatched DALI group in the Browser and must be mapped to real DALI interfaces and ballasts using this window.

Once you have discovered and addressed all the real DALI ballasts for each interface (if more than one) you can then patch your DALI fixtures to them simply by dragging them from the Unpatched DALI group onto the required interface and address cell. As each DALI interface is assigned DALI fixtures, the Browser refreshes to reflect these changes. The Unpatched DALI group will become empty once all the DALI fixtures in the project have been patched.

It is of course possible to patch your DALI fixtures blind in advance of being connected to the real ballasts. The patch is stored with the project data and not on the DALI ballasts.

DALI groups

Unlike other groups in Designer, DALI groups are a property of the real DALI interface not an abstract collection of fixtures, and there can only be 16 DALI groups per interface. The matrix on the lower pane of the window allows you to add and define groups for each interface:

To add a DALI group:

1. Press the Add button
2. Type a name by double-clicking the name cell

3. Select which ballasts are to be a member by double-clicking the ballast's cell

The DALI group is added to the Browser and group configuration data ready to be uploaded into the ballasts.

To delete a DALI group:

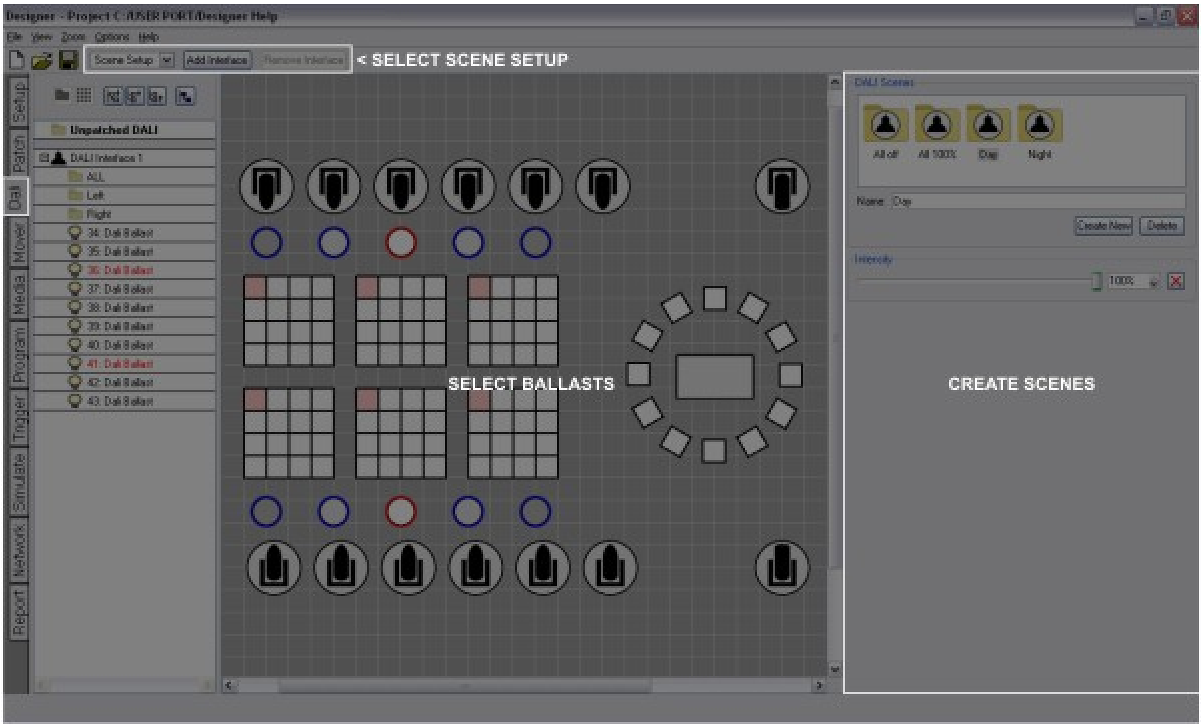
1. Right-click on the group in the Browser
2. Select Delete
3. Press Ok to confirm

The DALI group is removed and the group configuration data updated ready to be uploaded into the ballasts.

If groups have already been programmed onto the DALI fixtures you can press the Discover Groups button to automatically populate group information from the fixtures on the interface.

DALI scenes

In Designer's implementation, DALI scenes are common across DALI interfaces, use the pull-down on the DALI toolbar and select Scene Setup for programming:



The Browser and project plan are displayed so that you can select the DALI fixtures and groups. On the right are the DALI scenes folder and the Intensity controls.

There can be up to 16 DALI scenes which can contain programming for some or all of your DALI fixtures, even if spanning different DALI interfaces. In general you should include programming for all your DALI fixtures in each scene since you can determine when creating timelines which fixtures or groups should run the scene by dropping the DALI scene on the appropriate timeline row, the choice is yours though.

To create a DALI scene:

1. Press the Create New button
2. Name the new scene
3. Select the DALI fixtures or groups
4. Set the required level (0>254), the fixtures on the plan will simulate these levels

The DALI scene is added to the folder and scene configuration data is ready to be uploaded into the ballasts.

To delete a DALI scene:

1. Select the scene in the folder
2. Press Delete
3. Press Yes to confirm

The DALI scene is removed and scene configuration data updated ready to be uploaded into the ballasts.

To edit a DALI scene:

1. Select the scene in the folder
2. Select the DALI fixtures or groups
3. Adjust the levels

The DALI scene is edited and scene configuration data updated ready to be uploaded into the ballasts.

To remove a DALI fixture from a scene:

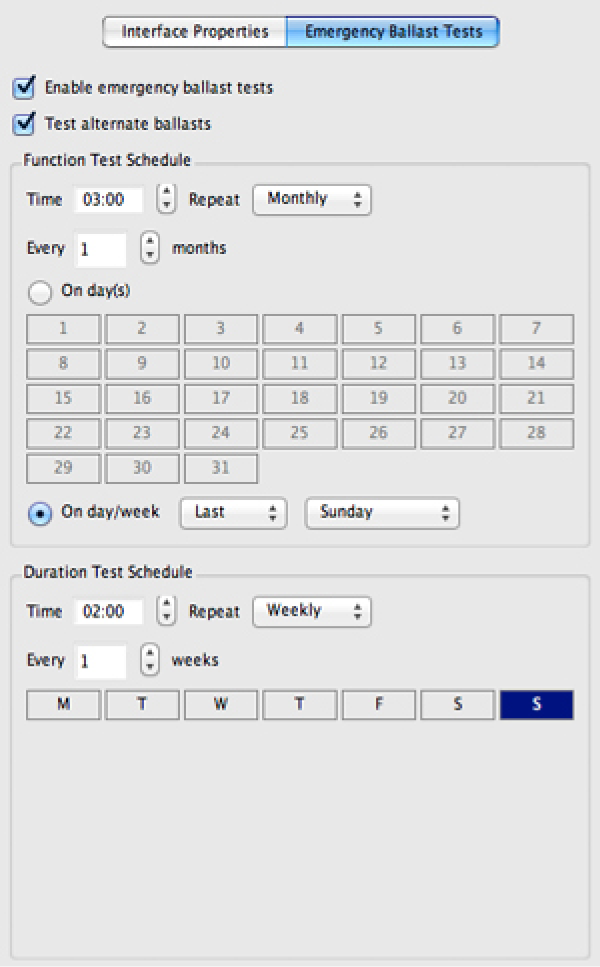
1. Select the scene in the folder
2. Select the fixture to remove
3. Press the ✕ Knockout button to the right of the Intensity controls

The DALI scene is edited and scene configuration data updated ready to be uploaded into the ballasts.

Emergency Ballasts

Emergency DALI ballasts are set-up in the same way as standard DALI ballasts and support Highlight and Re-Addressing.

Emergency tests may be scheduled for all emergency ballasts in a project via the Emergency Ballast Tests tab on the right of the DALI window. You can override project settings on a per interface basis by using the emergency settings in the Interface Properties tab. Function and Duration tests can be scheduled independently of each other on a daily, weekly, monthly or yearly basis. The time of day that the tests run can also be specified. Choosing to test alternate ballasts will test every other patched ballast - the remaining ballasts will then be tested once the initial test is complete.



☑ Enable emergency ballast tests

☑ Test alternate ballasts

Function Test Schedule

Time 03:00 Repeat Monthly

Every 1 months

○ On day(s)

● On day/week Last Sunday

Duration Test Schedule

Time 02:00 Repeat Weekly

Every 1 weeks

Test information is stored on the memory card of the controller responsible for that interface. This information can also be viewed via the web interface.

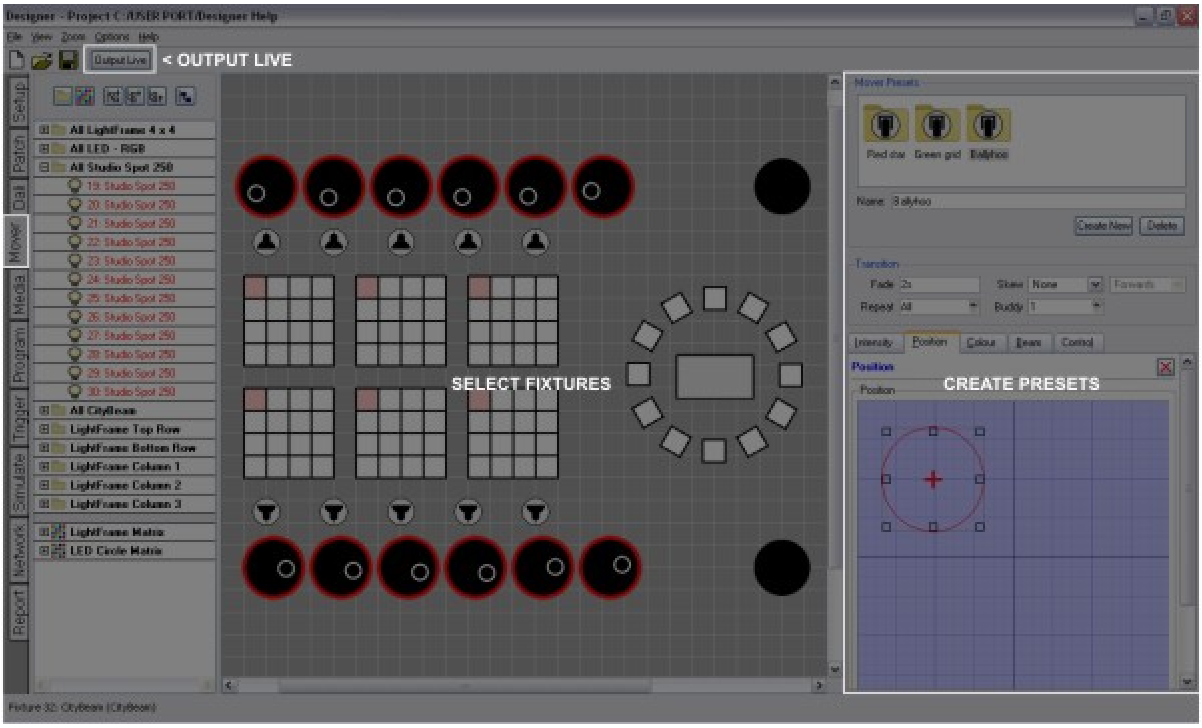
Specify a ballast's Prolong time in the Setup tab.

Upload configuration

Once you have configured all your DALI interfaces and programmed all your DALI scenes you must upload the configuration to each DALI interface in turn so that this data can be stored on the DALI ballasts themselves. Select an interface using the pull-down on the DALI toolbar and press the Upload Configuration button, a progress bar will track this rather slow procedure.

Mover

This window need only be used if there are fixtures in the project that have control parameters other than intensity and colour mixing. Examples include most automated lights which, above and beyond position (pan & tilt), might also include gobo and effects wheels:



Name: Ballyhoo

Fade: 2s

Snap: None

Repeat: All

Buddy: 1

Fixture 32: CityBeam (CityBeam)

The Mover window is where one sets these other parameters and it comprises four sections: On the left is the Browser, in the middle the Plan. Top right is the directory of Mover Presets within the project and below that is the Fixture Parameters pane and it is here that parameters other than intensity and colour mixing are set.

Parameter kinds

Parameters are grouped by kind:

Intensity	Parameters excluding intensity (which is programmed directly on the timeline) such as strobe
Position	Pan and tilt
Colour	Parameters excluding colour mixing (which is programmed directly on the timeline) such as fixed colour wheels and scrollers, colour correction
Beam Image	Gobo and effects wheels
Beam Shape	Iris, zoom and focus etc.
Control	Lamp and fixture controls such as dowse, strike, reset and macros

Setting one parameter within a kind sets or “touches” all the rest, the kind heading will turn blue and the Knockout button will be enabled to indicate that programming for this kind is included in the preset.

✕ To remove programming press the Knockout button to the right of the kind heading to “untouch” these parameters.

Effects

A realtime effect can also be applied to any parameter by pressing the associated FX button which opens the effect controls for adjustment, pressing FX again turns off that effect.

Effects work by modulating the programmed base value with a low-frequency oscillator (LFO):

- Effect - the shape of the LFO's waveform (Square, Triangle, Sawtooth, Sine or Cosine)
- Size - the amplitude of the modulation, adjust in conjunction with the base value
- Period - the number of seconds over which the waveform will repeat, the higher the period the slower the effect
- Style - choose None (all parameters are modulated together) or Spread (the effect is spread over space as well as time)

Position

The Position kind is a special case in that a) a graphical position picker is provided and b) the pan and tilt effects are linked and displayed/adjusted graphically. Hold Shift while dragging for fine resolution or use the Pan and Tilt fields provided underneath for accurate, numerical positioning. Note that you can also lock an axis by clicking on the Pan or Tilt button which allows positioning of one axis without interfering with the other.

Creating a mover preset

Click Create New to make a new preset and type a name. Select the fixtures, which will automatically come on to 100% so that you can see what you’re doing, set their parameters and add effects to these parameters as required. The fixture icons on the plan will provide graphical feedback of what they’re doing. You will not be able to select any fixtures unsuitable for making mover presets. Use the Transition section to give the preset a default fade time and skew type, the preset is now ready for use.

Mover Presets can be created to contain programming for all kinds (complete look) or just a single kind, for example Colour presets or Position presets, akin to palettes on a moving light console. This latter method allows for more flexibility when programming the timelines as you can mix and match these kind presets to make complete looks on the fly. It also allows different transitions to be applied to each kind so that gobos could fly in one after another (using a skew) while position moves more slowly and as a whole (longer fade, no skew).

Using Output Live

While the mover icons give a fair degree of feedback about your programming, by visualising colour and gobos for example, other parameters are not visualised and so must be programmed live, press the Output Live button on the main toolbar to do so. Output Live is not possible without patched and associated Controllers.

Note that if you wish to program or edit a preset that requires one or more other presets to be on stage, for example rotating, zooming and focusing a gobo where the position is set by another preset, then you must also be simulating a timeline that runs those presets.

Editing a mover preset

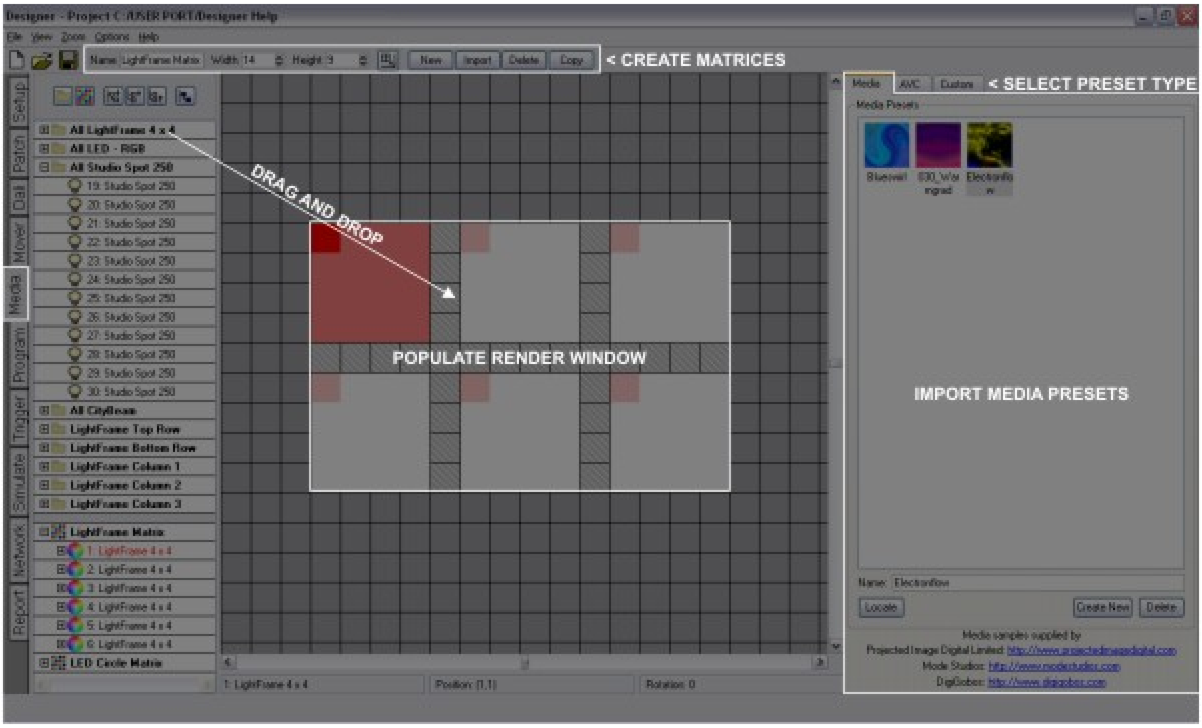
To edit an existing preset just click on it and select the fixtures for adjustment, note that included fixtures will be shown with a blue halo on the plan and fixtures currently selected for edit with a red halo. Any timelines on which the preset is deployed will automatically update to reflect the edits made.

Deleting and copying a mover preset

Right click on a preset to delete or copy it, the latter function being useful in making similar presets that vary only slightly, say just the gobo being used.

Media

This window serves two purposes: to edit or manually create Pixel Matrices of fixtures for use as a media playback screen, and to import Media, AVC & Custom Presets into the system:



The window comprises 3 sections: On the left is the Browser, in the middle the Pixel Matrix editing area, and on the right are the Presets panes.

Pixel Matrices

To create Pixel Matrices automatically from the plan (recommended):

1. Go to Setup
2. Select the fixtures you want to include in the Matrix
3. Press the New Pixel Matrix button on the Browser toolbar, the software will automatically create a Matrix with the fixtures correctly positioned and the Render Window cropped to best fit
4. Name the Pixel Matrix using either the textbox on the toolbar or right-clicking>Rename in the Browser

To create a Pixel Matrix manually:

1. Press Create New on the toolbar, a default 50x50 Render Window (the hatched area) will be created and the Browser populated
2. Populate the Render Window by dragging on fixtures from the Browser
3. Adjust the size of the Render Window by using the Width & Height fields on the toolbar or the Crop button (see below) to best fit
4. Name the Pixel Matrix using either the textbox on the toolbar or right-clicking > Rename in the Browser

Typically you will place fixtures to mimic as closely as possible their actual deployment, the software will compensate for gaps and irregularities in a matrix so that media will be rendered correctly. Fixtures that have been rotated on the Plan during Setup will be placed on the Matrix using this rotation although this is only relevant to compound fixtures. Such fixtures can be rotated separately for each Matrix by dragging them around when the cursor indicates rotate mode, note that the first element of a compound fixture is displayed a darker pink for easy orientation.

To remove a fixture from a Matrix right-click on the fixture on the plan and select Remove Fixture.

To break a composite fixture apart into individual pixels right-click on the fixture on the plan and select Break Fixture.

You may create as many Matrices as you like and any fixture can be used as a “pixel” in any Matrix but clearly those that can colour mix are the sensible choice, RGB LEDs being by far the best due to their large colour gamut and fast response. That being said, the software will render coloured media as grey scale on intensity only fixtures.

Crop render window

Use the Crop Render Window button to trim the Render Window to fit the extents of the fixture array.

Manual sizing of the Render Window is provided by typing in appropriate Width and Height size values and it is perfectly allowable to undersize the Render Window to achieve effects such as picture-in-picture or oversize it to concentrate on a particular area of the imported media.

Import pixel matrix

You can use the Import button to import pixel layout from a CAD application via a CSV file, see pixel matrix file format.

Media Presets

Select the Media tab. Media presets can only be played on MSC Pixel Matrices.

Media is imported by clicking Create New in the Media Presets pane. A dialog will open as shown above for you to browse to wherever your media files are located, enable Thumbnails view to preview them. Designer supports most common media files, indeed anything that QuickTime supports, and it matters not at all the resolution of the media clips as the software will up/down-scale as required when the Media Preset is placed on a Matrix during programming. In short, no media preparation should be required. The first frame is used as their thumbnail in the directory, they can be named using the name textbox at the bottom of the directory pane and they can also be deleted.

Imported Media is automatically resized to fit a Pixel Matrices’ Render Window, the hatched area of the Matrix. Only fixtures/pixels within this Render Window will receive the media which is why Render Windows should typically be cropped (see above) to force the media to fit onto just the fixtures.

Designer ships with sample media clips kindly provided by Projected Image Digital, Mode Studios and Digigobos. Four samples are installed with the application, the others reside in the Sample Media folder on the CD. Links are provided from the Media window to peruse these vendors’ websites to obtain additional media for your project.

Media Slots

Select the Media Slots tab. Media slots are placeholders for Media Presets and can only be played on MSC Pixel Matrices.

Create a Media Slot by clicking Create New in the Media Slots pane. A Media Slot can be renamed using the name text box underneath the Media Slots directory. The selected Media Slot can be deleted by clicking the Delete button.

The content for Media Slots is created using a separate utility application: Unison Mosaic MediaManager. When you have finished creating your Designer project, go to File > Export for Media Manager to save details about your Media Slots to a .mdx file. Create a new project in MediaManager and import the .mdx file. You can now program content onto each media slot. It is possible to import .mdx files from multiple Designer projects into a single MediaManager project in order to aid media management for large installations. Please refer to the MediaManager help for further details.

Whenever you make a change to the fixtures, patch or programming in your Designer project you must go to File > Export for Media Manager and then import the generated file into your existing MediaManager project.

When you've exported your media from MediaManager you can upload it to your controllers using Unison Mosaic RemoteManager. Please refer to the RemoteManager help for further details.

AVC Presets

Select the AVC tab. AVC presets can only be played on AVCs, see Working with the Audio Visual Controller.

You must prepare the video and audio clips you want to use on the AVC in this pane. To add a new clip click Create New and browse to a suitable media file containing video, audio or both.

Should the media file(s) subsequently be moved, renamed or deleted such that the preset path(s) no longer valid then a warning icon will be displayed over the thumbnail.

Designer can import a wide variety of video formats and convert them into the video format used on the AVC. However, the AVC format is based on MPEG2 and for best quality results - and fastest conversion - we recommend preparing content as standard definition MPEG2. All video content must have either PAL (25fps) or NTSC (29.97fps) frame rates - Designer will not accept content with other frame rates, but it will scale video prepared for different resolutions.

Designer can also import a wide range of audio formats including AC3 (used on DVDs), WAV, and MP3. To add audio to an existing clip, make sure the clip is selected and click on the button next to the "Audio Path" box. You can then browse to the audio file and select it.

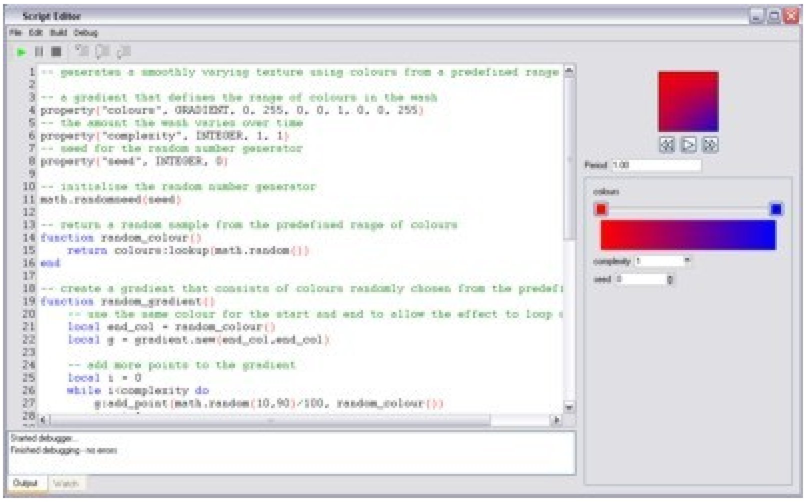
If the video and audio need to be synchronised (i.e. audio is a soundtrack for the video) then you should select "Audio synced" to ensure they are always locked together. You can specify an audio offset to correct for any time delay between video and audio. If the audio and video can run independently (e.g. audio is a music track of a different length to the video) then make sure "audio synced" is not selected and they will loop independently.

Custom Presets

Select the Custom tab. Custom presets can only be played on MSC Pixel Matrices.

Custom Presets use a Lua script to define an effect that can be played back on a Matrix. You can use this to create effects that are not available as standard in Designer.

To create a new Custom Preset, press Create New in the Custom tab on the right. This will open a script editor dialog:



```
-- generates a smoothly varying texture using colours from a predefined range

-- a gradient that defines the range of colours in the mesh
property("colours", GRADIENT, 0, 255, 0, 0, 1, 0, 0, 255)
-- the amount the mesh varies over time
property("complexity", INTEGER, 1, 1)
-- seed for the random number generator
property("seed", INTEGER, 0)

-- initialise the random number generator
math.randomseed(seed)

-- return a random sample from the predefined range of colours
function random_colour()
    return colours:lookup(math.random())
end

-- create a gradient that consists of colours randomly chosen from the predefi
function random_gradient()
    -- use the same colour for the start and end to allow the effect to loop s
    local end_col = random_colour()
    local g = gradient.new(end_col,end_col)

    -- add more points to the gradient
    local i = 0
    while i<complexity do
        g:add_point(math.random(10,90)/100, random_colour())
```

```
Started debugger...
Finished debugging - no errors
```

The source editor initially shows the framework of a custom preset. You can either enter the source yourself, or you can load the source from a file using File > Import. Designer provides sample scripts which are located in Program Files at \ETC\Mosaic\Designer\resources\scripts\custom_presets.

After editing or importing a script, you compile it by pressing F7. If there are any errors in the script, they will be reported in the Output tab. If there are no errors reported, it is advisable to run the script to preview the preset. To run the script, use Debug > Run (F5). If the script takes a long time to run, or appears to have got stuck in a loop, use Debug > Stop (Ctrl+F5) to stop execution.

IMPORTANT: When working in the script editor, if your script does have an infinite loop, you are able to stop it executing by pressing Debug > Stop (Ctrl+F5). However, outside of the debugger, there is no such way to stop the execution of a badly-behaved script. This will result in Designer locking up and will have to be shut down manually. You must ensure that your scripts do not have such errors in them.

A preview of the preset will be generated and shown on the right of the script editor. You can start and stop the preview and step forwards and backwards with transport controls below the preview. Altering the period below the preview and pressing Debug > Run (F5) again will generate the preview again with the specified period.

If the preset defines any properties, after the script is successfully compiled, suitable editors for these properties will be displayed below the preview. You can change the values of these properties and rerun the script using Debug > Run (F5) to observe the effect of those properties in the preview.

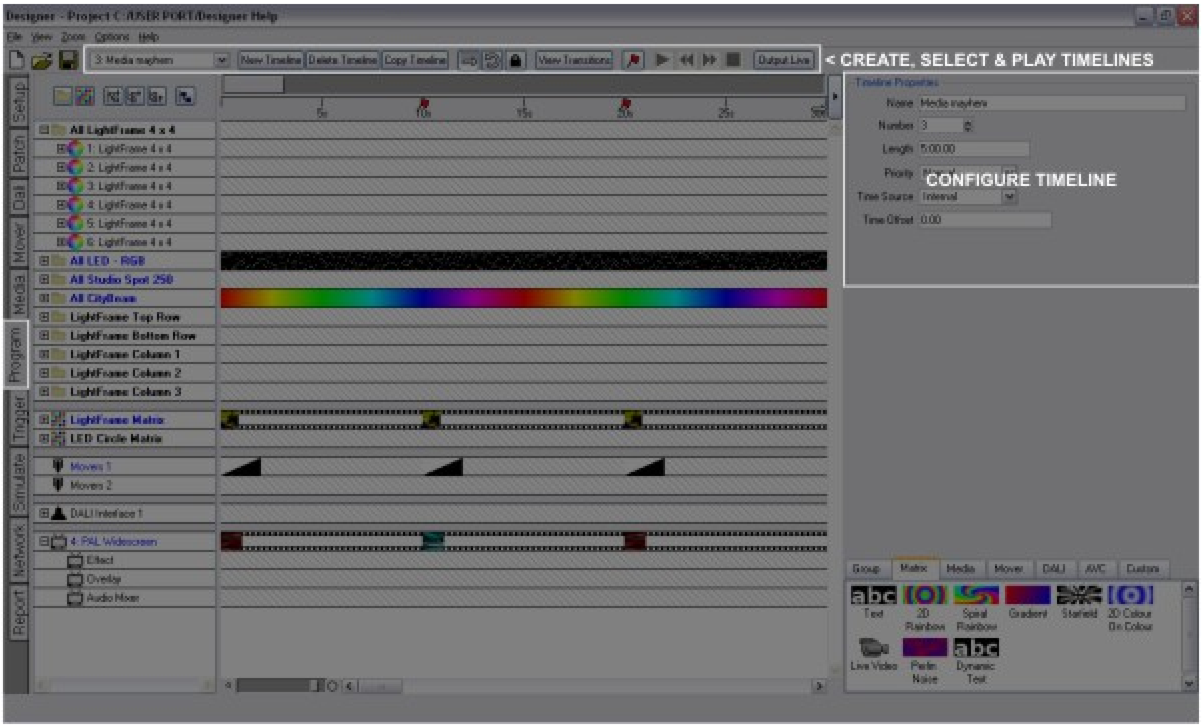
Once you are happy with the preview, close the script editor. The preset can now be placed on Matrices on a timeline.

You can edit the source of the custom preset script again by selecting it in the Custom tab on the Media pane and pressing Edit. If you edit a preset that is already used on a timeline, any changes you make will be applied to everywhere where the preset is used.

Refer to the script editor documentation for more information on editing scripts, and the custom presets programmers guide for help on creating a custom preset from scratch.

Program - Timeline Properties

With no presets selected, the Timeline Properties pane is displayed:



< CREATE, SELECT & PLAY TIMELINES

Timeline Properties

Name: Media mayhem

Number: 3

Length: 5:00.00

Priority: Normal

Time Source: Internal

Time Offset: 0.00

CONFIGURE TIMELINE

Name

Give your timeline a name here, a descriptive name will help you identify the correct timeline when creating basic triggers and viewing the web interface's status and control pages.

Number

Every timeline has a unique number which is primarily for reference but can be changed if necessary. The timeline number is used to identify a timeline for creating advanced triggers, for example when using LUA scripts, and when using the web interface's command line.

Length

The default timeline length is 05:00.00 (5 minutes) and you will need to increase this before placing or extending presets beyond this time. The maximum timeline length is 24 hours to prevent them becoming unmanageable - use triggers to stitch together multiple timelines to create longer time frames.

Priority

Use the pull-down to select a priority level for the timeline:

- High - the timeline will never be overridden
- Above normal
- Normal - the timeline behaves normally and can be overridden
- Below normal
- Low - the timeline can not override other timelines

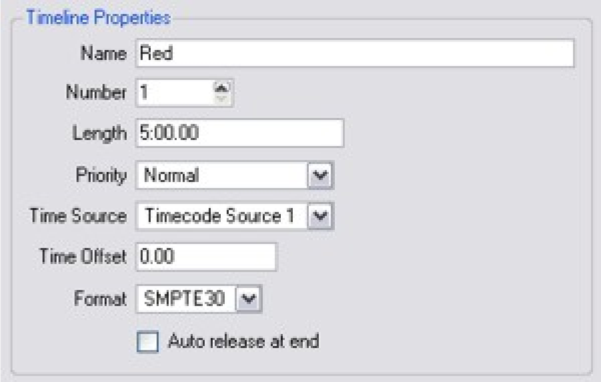
Time source

Use the pull-down to select a time source for the timeline to follow:

- Internal - the timeline will run autonomously although playback speed and position can be overridden using triggers.
- Timecode Bus - the timeline will follow one of six Timecode Buses (see below).
- Audio Bus - the timeline will follow one of four Audio Buses (see below).

Working with timecode

By selecting one of the six Timecode Buses, the timeline's ruler will display timecode values and the properties pane will give further options:



Timeline Properties

Name: Red

Number: 1

Length: 5:00.00

Priority: Normal

Time Source: Timecode Source 1

Time Offset: 0.00

Format: SMPTE30

□ Auto release at end

Time Offset

Timelines by default start at 00:00:00.00 (hours:minutes:seconds.frames) but the timecode source may not do so, the tape may have been "striped" with an offset of an hour (01:00:00.00) for example. Enter the source's starting value in this box to synchronise.

Format

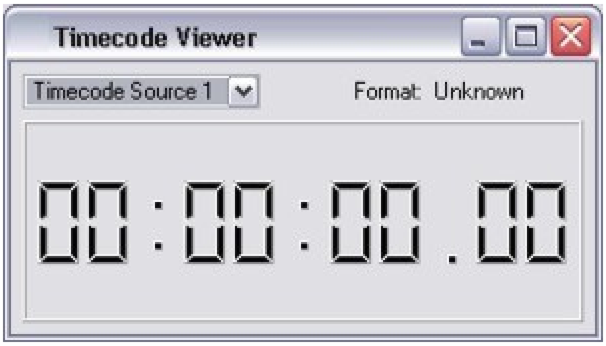
Timecode comes in four formats that depend on the source media used, select the appropriate format here (Film24, EBU25, SMPTE30 & NTSC30) to prevent missed frames and stuttering playback.

Auto release at end

Check this box if you want the timeline to release, and hence ignore timecode, at the end (the end of the last placed preset or flag). Typically leave it unchecked so that the timeline always follows the timecode regardless of the timecode value which may exceed the end time.

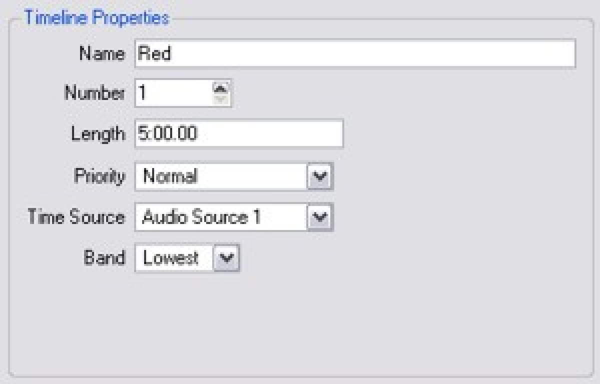
Timecode Buses

The six Timecode Buses are internal buses to which one patches the external timecode sources available to the system. These may be MIDI timecode (MTC) sources input via one or more MSCs' or RIO As' MIDI Inputs or linear timecode (LTC) sources input via one or more RIO As or LTC Expansion Modules. You can use the Timecode Viewer available from the View menu to monitor each Timecode Bus:



Working with audio

By selecting one of the four Audio Buses, the properties pane will give further options:


Timeline Properties

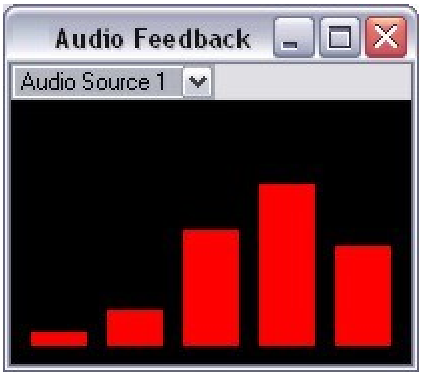
Name: Red
Number: 1
Length: 5:00.00
Priority: Normal
Time Source: Audio Source 1
Band: Lowest

Band

The Audio Expansion Module generates five frequency bands (63Hz, 250Hz, 1KHz, 3.5KHz and 12KHz) and the RIO A can generate up to 30 frequency bands (configured in Remote Device properties). Use this pull-down to select which band will drive the timeline.

Audio Buses

The four Audio Buses are internal buses to which one patches the external audio sources available to the system via one or more RIO As or Audio Expansion Modules. You can use the Audio Feedback window available from the View menu to monitor each Audio Bus:

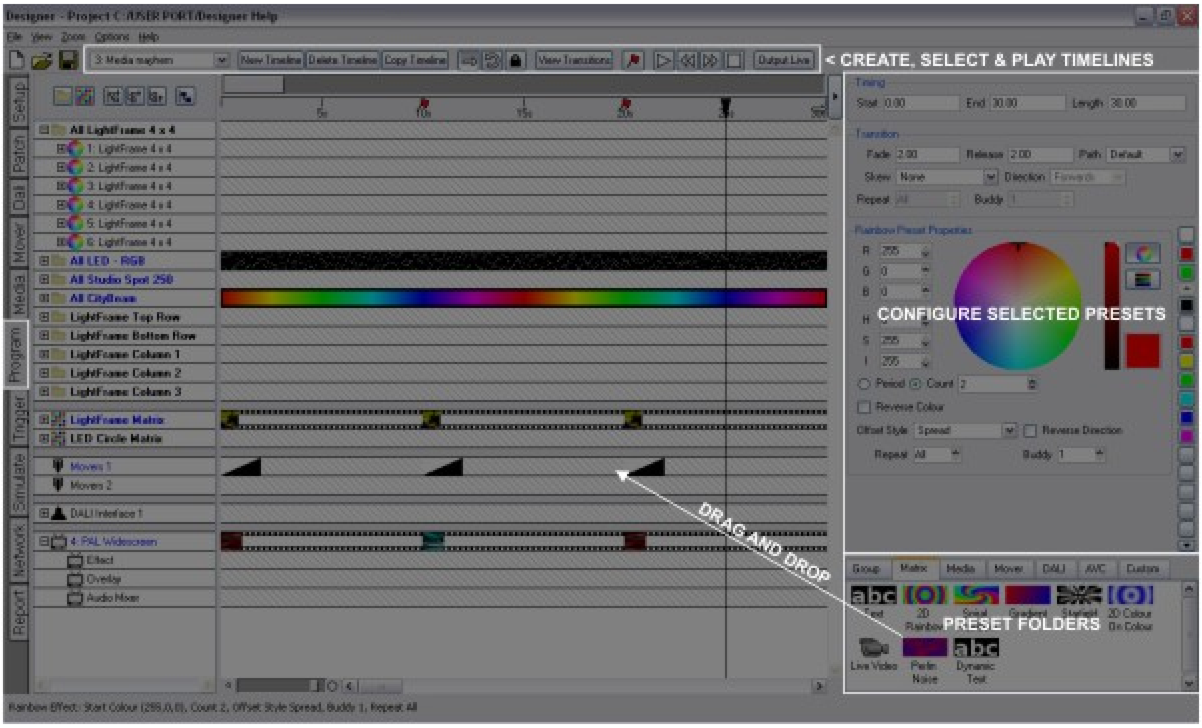


Changing the timeline & preset defaults

Use Options > Preferences on the main toolbar and select the Timelines tab to change these defaults. Here you can change the default timeline name and length as well as the length, fade & release time and default crossfade path of newly placed presets. See Preferences.

Program - Preset types & properties

The Program window is where you put your presentation together by dragging and dropping the built-in effects and your Mover, Media and Custom Presets onto your fixtures, groups and Pixel Matrices:



< CREATE, SELECT & PLAY TIMELINES

CONFIGURE SELECTED PRESETS

DRAG AND DROP

PRESET FOLDERS

Rainbow Effect: Start Colour (255,0,0), Count 2, Offset Style Spread, Buddy 1, Repeat All

The window comprises 4 sections: On the left is the Browser, in the middle the Timeline editing area with its associated toolbar across the top. Bottom right are the folders of Group, Matrix, Media, Mover, DALI, AVC and Custom Presets (although not all folders may be displayed). Above this is the Preset Properties pane which is divided into Timing, Transition and Properties all of which you use to manipulate how a preset placed on a timeline is rendered.

It is useful simultaneously to view the output of a timeline to check your programming thus the Simulate window can be opened at the same time as the Program window by selecting View > Tear off simulate or by holding Ctrl (Apple) and pressing F8, see Simulate for details.

Before creating a timeline it is worth covering the seven preset types:

Group Presets

A range of presets from simple colour fills to rainbow, sparkle and chase effects. These presets have no spatial awareness and are applied to fixtures and groups using the fixture ordering (as set in the Browser) to determine how they are rendered and in what order the fixtures will transition. They are akin to the sort of programming, effects and cue timing that is available on moving light consoles:

🟥 Fixed colour

The most basic preset, renders a static colour fill. Use the colour picker, user palette or text entry fields (RGB or HSI) to select the colour.

Colour fan

Renders a static multi-colour gradient over a group of fixtures:

- To change a colour, press on the coloured button, select a colour and press Ok
- To move a colour, click and drag the coloured button
- To add a new colour, click anywhere on the slider where there is no button
- To remove a colour, right-click on the coloured button
- Repeat - the number of elements between the start and end of the fan
- Buddy - the number of elements that will be set to the same colour in the fan (if Buddy is greater than 1, the number of elements that are repeated over is Repeat multiplied by Buddy)

Random colour

Renders a dynamic chase through a random sequence of colours:

- Colour - specifies the first colour of the sequence, all subsequent colours are relative to the start colour in a pseudo-random way (saturation and intensity levels are maintained)
- Period - the number of seconds that the sequence takes to complete
- Count - the number of times that the sequence should complete over the length of the preset
- Offset Style - choose None (all elements are the same colour) or Spread (the sequence is spread over space as well as time)
- Fade - the fade time in seconds between each colour in the sequence
- Hold - the time that each colour in the sequence is not fading
- Steps - the number of steps in the sequence
- Seed - the seed of the pseudo-random sequence (copying this value to another preset will create the same random sequence)
- Randomise - picks a new seed

Rainbow effect

Renders a dynamic rainbow effect cycling through hue:

- Colour - specifies the start colour of the rainbow (the saturation and intensity are maintained throughout the cycle)
- Period - the number of seconds that the rainbow takes to complete one cycle
- Count - the number of times that the rainbow should cycle over the length of the preset
- Reverse Colour - reverses the direction around the hue circle (default (unchecked) is clockwise)
- Offset Style - Choose None (all elements are the same colour), Spread (the rainbow is spread over space as well as time) or One Shot (as None, but will stop after one cycle of the hue circle)
- Reverse Direction - if Offset Style is Spread, reverse the direction of the spread in space
- Repeat - the number of elements between the start and end of the hue circle
- Buddy - the number of elements that will be set to the same colour in the rainbow (if Buddy is greater than 1, the number of elements that are repeated over is Repeat multiplied by Buddy)

Strobe

Renders a dynamic colour strobe effect on black:

- Transparency - select Opaque for none, Base Transparent to superimpose the effect onto other programming

- Colour - specifies the flash colour
- Period - the interval in seconds between the start of each flash
- Duration - the length in seconds of the flash

Sparkle

Renders a dynamic random sparkle effect:

- Base Colour - the colour of the background
- Spark Colour - the colour of the spark
- Transparency - select Opaque for none, Base or Spark Transparent to superimpose the effect onto other programming
- Period - the rate of the effect (larger numbers are slower)
- Density - the density of the effect in percent (higher numbers, more sparks)

Colour chase

Renders a dynamic chase through a user-specified sequence of colours:

- Colours - pressing the colour buttons will prompt for a new colour for that step
- Steps - the number of steps in the sequence
- Direction - choose Forwards, Backwards or Bounce (the latter uses two periods to complete)
- Period - the number of seconds that the sequence takes to complete
- Count - the number of times that the sequence should complete over the length of the preset
- Offset Style - choose None (all elements are the same colour) or Spread (the sequence is spread over space as well as time)
- Repeat - the number of elements to repeat the chase over
- Buddy - the number of elements that will be set to the same step (if Buddy is greater than 1, the number of elements that are repeated over is Repeat multiplied by Buddy)
- Fade - the fade time in seconds between each colour in the sequence
- Hold - the time that each colour in the sequence is not fading

Colour on colour

Renders a dynamic pulse of colour passing over another colour:

- Base Colour - the base colour
- Top colour - the colour of the pulse
- Transparency - select Opaque for none, Base or Top Transparent to superimpose the effect onto other programming
- Period - the number of seconds that the sequence takes to complete
- Count - the number of times that the sequence should complete over the length of the preset
- Repeat - the number of elements to repeat the pulse over
- Buddy - the number of elements that will be set to the same colour in the pulse (if Buddy is greater than 1, the number of elements that are repeated over is Repeat multiplied by Buddy)
- Shape - the shape of the pulse (Sine, Triangle, Square, Ramp Up or Ramp Down)
- Pulse Width - the width of the pulse in percent (1 > 200%, if 100%, the pulse is half of the element width)
- Phase - the offset of the pulse in degrees
- Reverse Direction - reverses the direction of the pulse
- Invert Pulse - changes the starting position of the pulse

Intensity

Renders static or dynamic intensity control:

- Level - intensity level
- Shape - choose None (static intensity) or a dynamic effect (Sine, Cosine, Square, Triangle or Ramp Up)
- Size - the amplitude of the effect
- Period - the period of the effect in seconds
- Count - the number of times that the effect should repeat over the length of the preset
- Offset Style - choose None (all elements are the same intensity), Spread (the effect is spread over space as well as time) or Once (as None, but will stop after one period)
- Reverse Direction - reverses the direction of the effect
- Repeat - the number of elements to repeat the effect over
- Buddy - the number of elements that will be set to the same intensity (if Buddy is greater than 1, the number of elements that are repeated over is Repeat multiplied by Buddy)

Flicker

Renders dynamic,random flickering over a colour gradient:

- To change a colour, press on the coloured button, select a colour and press Ok
- To move a colour, click and drag the coloured button
- To add a new colour, click anywhere on the slider where there is no button
- To remove a colour, right-click on the coloured button
- Period - the period of the effect in seconds
- Sub - the amplitude of the low frequency perturbation
- First - the amplitude of the fundamental flicker frequency
- Second - the amplitude of the second harmonic
- Third - the amplitude of the third harmonic
- Seed - used to offset the effect; click the Randomise button to generate a random value
- Uniform - apply the Seed value as the offset for all fixtures in the group, or use it as a seed to generate random offsets for each fixture in the group

Each of the sliders corresponds to a sine wave of a specific frequency. The frequency of Sub is defined by the Period - the default is 30 seconds (1/30Hz). First will be twice as fast, Second twice as fast again and and Third twice as fast as Second. The value of each sine wave is used to fetch a value from a set of pre-generated random values and the four results are summed. The sum is used to select a position in the colour gradient to output to the fixtures. Mix the different frequency components using the sliders to select how much of each component you want.

So if you are looking for a relatively steady flicker you might have a lot of Sub, with a little bit of Third to stop it looking too regular. If you want a more chaotic looking flicker then you might have less of Sub and First and more of Second and Third. It really is something you have to experiment with. If you want the overall flicker to have a different speed change the Period and everything will shift accordingly.

If you've got a set of slider values that you like and you want to copy the effect to another group, but not have both groups flickering identically, then just click the Randomise button to change the offset.

Hue Fade

Performs a fade in hue between two defined points:

- Start Colour - defines the hue at the start of the preset, and the saturation and brightness throughout the preset
- End Colour - defines the hue at the end of the preset; saturation and brightness will be the same as the start colour
- Reverse Direction - reverses the direction of the hue fade

The start and end colours will share the same saturation and brightness; editing the saturation or brightness for one colour will edit the other as well.

The fade time between the colours is determined by the length of the preset on the timeline.

Matrix Presets

A range of presets from gradient colour fills to rainbow and text effects. These presets have spatial awareness and can be applied to Pixel Matrices using the inherent 2D information to determine how they are rendered and in what order the fixtures will transition. They are considerably more powerful than the standard presets and allow you to “paint” onto your fixture arrays:

Text

Renders a scrolling text message on a matrix:

- Text Colour - the text colour
- Base colour - the background colour
- Period - the number of seconds that the message takes to scroll over the matrix
- Count - the number of times that the message should scroll over the length of the preset
- Text - the text to render
- Offset - the percentage of the height of the matrix to offset the baseline of the text (positive is down, negative is up)
- Text Height - the height of the text in percentage of the matrix height
- Direction - the direction to scroll in
- Flip - flips the text upside down
- Mirror - flips the text left to right
- Bold - renders the text in bold
- Italic - renders the text in italic
- Font - the font to render the text in

2D rainbow

Renders a dynamic rainbow effect on a matrix:

- Colour - specifies the start colour of the rainbow (the saturation and intensity are maintained throughout the cycle)
- Period - the number of seconds that the rainbow takes to complete one cycle
- Count - the number of times that the rainbow should cycle over the length of the preset
- Type - the shape of the rainbow effect (Linear, Radial, Conical, Square, Noise, Perlin Noise or Bilinear)
- Reverse - reverses the direction of the wave

If the Type is Linear, Radial, Conical, Square or Bilinear, the properties Angle, Repeat and Count are available:

- Repeat - the repeat style (None, Sawtooth, Triangle)
- Count - the number of repeats
- Angle - the angle in degrees of the wave (Linear, Conical & Bilinear only)

Note that setting Repeat to None will only have an apparent effect when the Type is Radial. It behaves like Sawtooth with a Count of 1, except that the area outside the unit circle is filled with the same colour as the edge of the unit circle, rather than the effect continuing beyond a Count of 1.

If the type is Noise:

- Seed - the seed of the pseudo-random noise (copying this value to another preset will create the same noise)
- Randomise - picks a new seed

If the type is Perlin Noise:

- Seed - the seed of the pseudo-random noise (copying this value to another preset will create the same noise)
- Randomise - picks a new seed
- X - the horizontal scale (higher numbers will produce more variation horizontally)
- Y - the vertical scale (higher numbers will produce more variation vertically)
- Depth - the coarseness of the noise

Spiral rainbow

A special version of 2D Rainbow with conical type:

- Colour - specifies the start colour of the rainbow (the saturation and intensity are maintained throughout the cycle)
- Period - the number of seconds that the rainbow takes to complete one cycle
- Count - the number of times that the rainbow should cycle over the length of the preset
- Spokes - the number of times the rainbow repeats around the matrix
- Curve - an angle in degrees to curve the effect as it radiates from the centre of the matrix
- Reverse - reverses the direction of the spiral

Gradient

Renders a static multi-colour gradient on a matrix:

- To change a colour, press on the coloured button, select a colour and press Ok
- To move a colour, click and drag the coloured button
- To add a new colour, click anywhere on the slider where there is no button
- To remove a colour, right-click on the coloured button
- Type - the shape of the rainbow effect (Linear, Radial, Conical, Square, Noise, Perlin Noise or Bilinear)

If the Type is Linear, Radial, Conical, Square or Bilinear, the properties Angle, Repeat and Count are available:

- Repeat - the repeat style (None, Sawtooth, Triangle)
- Count - the number of repeats
- Angle - the angle in degrees of the gradient (Linear, Conical & Bilinear only)

If the type is Noise:

- Seed - the seed of the pseudo-random noise (copying this value to another preset will create the same noise)
- Randomise - picks a new seed

If the type is Perlin Noise:

- Seed - the seed of the pseudo-random noise (copying this value to another preset will create the same noise)
- Randomise - picks a new seed
- X - the horizontal scale (higher numbers will produce more variation horizontally)
- Y - the vertical scale (higher numbers will produce more variation vertically)
- Depth - the coarseness of the noise

Starfield

Renders a radiating star field:

- Space Colour - the colour of the background
- Star Colour - the colour of the stars
- Speed - the speed of the stars
- Star Count - the number of stars to show

2D colour on colour

Renders a dynamic pulse of colour over another colour on a matrix:

- Base Colour - the base colour
- Top colour - the colour of the pulse
- Transparency - select Opaque for none, Base or Top Transparent to superimpose the effect onto other programming
- Period - the number of seconds that the sequence takes to complete
- Count - the number of times that the sequence should complete over the length of the preset
- Type - the type of pulse (Linear, Radial, Conical, Square, Noise, Perlin Noise or Bilinear)
- Shape - the shape of the pulse (Sine, Triangle, Square, Ramp Up or Ramp Down)
- Pulse Width - the width of the pulse in percent (1 > 200%, if 100% the pulse fills half of the matrix)
- Reverse Direction - reverses the direction of the pulse
- Invert Pulse - changes the starting position of the pulse

If the Type is Linear, Radial, Conical, Square or Bilinear, the properties Angle, Repeat and Count are available:

- Repeat - the repeat style (None, Sawtooth, Triangle)
- Count - the number of repeats
- Angle - the angle in degrees of the pulse (Linear, Conical & Bilinear only)

If the type is Noise:

- Seed - the seed of the pseudo-random noise (copying this value to another preset will create the same noise)
- Randomise - picks a new seed

If the type is Perlin Noise:

- Seed - the seed of the pseudo-random noise (copying this value to another preset will create the same noise)
- Randomise - picks a new seed
- X - the horizontal scale (higher numbers will produce more variation horizontally)
- Y - the vertical scale (higher numbers will produce more variation vertically)
- Depth - the coarseness of the noise

Live video

Displays live video on a matrix (MSC X only):

- X&Y offset - map to the top left pixel of interest on the incoming DV image

Perlin noise

Renders a smoothly-varying noise effect:

- To change a colour, press on the coloured button, select a colour and press Ok
- To move a colour, click and drag the coloured button
- To add a new colour, click anywhere on the slider where there is no button
- To remove a colour, right-click on the coloured button
- Period - the number of seconds that the noise takes to loop
- Count - the number of times that the noise should loop over the length of the preset
- X - the horizontal scale (higher numbers will produce more variation horizontally)
- Y - the vertical scale (higher numbers will produce more variation vertically)
- Depth - the coarseness of the noise
- Rate - the rate at which the noise varies
- Seed - the seed of the pseudo-random noise (copying this value to another preset will create the same noise)
- Randomise - picks a new seed

Dynamic text

Renders a text message which can be changed at runtime:

- Colour - the colour of the text
- Period - the number of seconds that the message takes to scroll over the matrix
- Count - the number of times that the message should scroll over the length of the preset
- Text - the text to render
- Font - the font to use to render the text (see the Fonts dialog below)
- Scroll - the scroll direction of the text
- Seamless Loop - if the text is set to scroll, setting this will remove the gap between the end and the start
- Alignment - if the text is set to not scroll, this is the alignment of the text (Left, Centre, Right)
- Mirror - flips the text left to right
- Flip - flips the text top to bottom
- Invert Colour - sets either the background or the text to be transparent
- Offset - set a offset amount on the Y axis of the matrix

To configure the font used by the Dynamic Text preset, press the [...] button next to the font picker to open the Fonts dialog:

- Select a font from the Font picker
- Press New to create a new font
- Press Delete to delete the selected font (note that you cannot delete a font that is in use in the project)
- Set the font's name in the Name property
- Use Family, Size, Bold and Italic to set the appearance of the font
- Press Ok to close the Fonts dialog

NOTE: Editing a font will change all Dynamic text presets that use that font not just the currently selected preset (s).

The Dynamic Text preset allows you to change the text after uploading the project to a Controller. To do this, you need to specify which parts of the text are going to change and which parts will remain the same.

For example, to show the opening time of a venue, you might set the Text property to "Opening Time: <open>". This creates a text slot called 'open' which you can change the value of. You can have more than one slot specified in the Text property, for example "Opening Time: <open> Closing Time: <close>".

To set the initial text for a text slot, press the [...] button next to the Text property to open the Dynamic Text Configuration dialog:

- Click in the Default Value cell of a slot to edit the text stored in that slot
- You can remove unused text slots by pressing Remove
- Press Ok to save changes and Cancel to discard changes

The Set Text Slot trigger action allows you to change the value of a text slot from a trigger.

There are two built-in slots, <time> and <date>, which show the current time and date respectively. You can change the format of how the time and date are displayed in the Dynamic Text Configuration dialog. Press the [...] button next to the Text property to open this dialog. At the bottom of the dialog you can select from some standard time and date formats, or type your own using the following codes:

%a	Abbreviated weekday name
%A	Full weekday name
%b	Abbreviated month name
%B	Full month name
%c	Date and time representation
%d	Day of the month (01-31)
%H	Hour in 24h format (00-23)
%I	Hour in 12h format (01-12)
%j	Day of the year (001-366)
%m	Month as a decimal number (01-12)
%M	Minute (00-59)
%p	AM or PM designation
%S	Second (00-61)
%U	Week number with the first Sunday as the first day of week one (00-53)
%w	Weekday as a decimal number with Sunday as 0 (0-6)
%W	Week number with the first Monday as the first day of week one (00-53)
%x	Date representation
%X	Time representation
%y	Year, last two digits (00-99)
%Y	Year
%Z	Timezone name or abbreviation
%%	A % sign

All other text is used verbatim. The computed output will be truncated to 255 characters.

Media Presets

Preset (user named)

The presets that you optionally created using the Media window to import still and moving images into your project. These presets have spatial awareness when applied to Pixel Matrices in that the media clip will be resized to fit the Pixel Matrix’s Render Window.

NOTE: If no Media presets have been created, the Media Presets folder will not be displayed.

Media Slot Presets

The Media Slots that you optionally created using the Media window. These presets have spatial awareness when applied to Pixel Matrices in that Unison Mosaic MediaManager will generate media for the Media Slot at the dimensions of the Pixel Matrix's render window.

Mover Presets

Mover presets do not have a length, only a transition (shown on the timeline), with the settings persisting until another Mover preset is encountered or the timeline released. They are applied to the dedicated Movers rows of a timeline.

Preset (user named)

The presets that you optionally created using the Mover window to control the other parameters, typically position and gobo, of any automated lighting in your project.

NOTE: If no Mover presets have been created, the Mover Presets folder will not be displayed.

DALI Presets

Like Mover presets, DALI presets do not have a length, only a transition, with the settings persisting until another DALI preset is encountered.

However, unlike Mover presets, DALI presets will persist even if the timeline is released. Indeed, since they are just commands to tell the DALI ballasts what to do, even power-cycling the Controller will make no difference; the settings will persist until a new command is issued or the ballasts themselves power-cycled.

Set level

Used to set a DALI fixture or user created group to a level (0>254, 255), and select a fade time from the pull-down list of DALI fade times. See DALI regards creating DALI groups.

Scene (user named)

Used to recall a DALI scene that you created and uploaded, and select a fade time from the pull-down list of DALI fade times. See DALI regards creating DALI scenes.

NOTE: If there are no DALI fixtures in the project, the DALI Presets folder will not be displayed.

AVC Presets

See Working with the Audio Visual Controller.

NOTE: If there are no AV fixtures in the project, the AVC Presets folder will not be displayed.

Custom Presets

{...} Preset (user named)

Renders a custom preset that you have optionally created using the Media window:

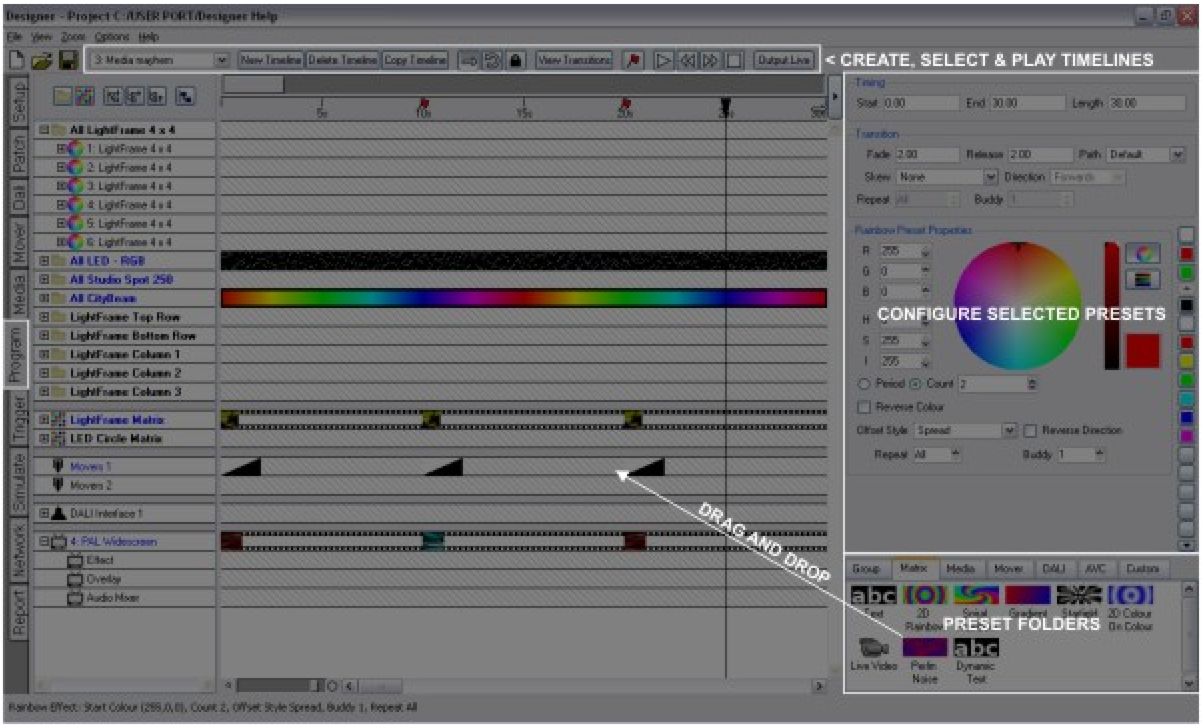
- Period - the number of seconds that the effect takes to complete one cycle
- Count - the number of times that the effect should cycle over the length of the preset

In addition, custom presets may define a number of properties that can be set for each instance of that preset on the timeline.

NOTE: If no Custom presets have been created, the Custom Presets folder will not be displayed.

Program - Working with timelines

Now that we've taken a look at the preset types let's see how to apply them to create a timeline:



Rainbow Effect: Start Colour (255,0,0), Count 2, Offset Style Spread, Buddy 1, Repeat All

Creating a timeline

To create a timeline click New Timeline and a fresh set of blank rows will appear to the right of each Browser entry. Use the Timeline Properties pane to name the timeline, the default length of which is 5 minutes which can be changed using the Length text box as required (timeline length is really just a user interface setting and can generally be left at the default value).

Timeline row categories

There are five categories of timeline row which determine the preset type that can be deployed on them:

Groups and fixtures

The majority of the timeline rows will be your groups of fixtures, the all groups created by the system as you added fixtures and the groups you made to organise your programming. Only Group presets can be applied to this category. Click the plus sign to expand a group and expose its members:

A fixture, or element within a compound fixture, capable of colour mixing, for example an RGB LED or automated light with CMY colour mixing. Use the Group colour presets to set static or dynamic intensity and colour.

A fixture incapable of colour mixing, for example a conventional light with/without a scroller or automated light with only a colour wheel. Use the Group intensity preset to set static or dynamic intensity. Create Mover presets to control the scroller or colour wheel as required.

Matrices

These are the Pixel Matrices created in Setup and Media, only Matrix, Media and Custom presets can be applied to this category. In many ways, creating matrices and working with these powerful presets is the preferred way to go with Designer.

Movers

Unlike the categories above these rows do not specify a fixture selection but are instead simply a placeholder for any Mover presets created in the project. The fixture selection is a property of the Mover preset depending as it does on the fixtures selected when the preset was created. More rows can be added as required by right-clicking in the Browser and selecting New Movers Row, similarly to remove them.

DALI

These rows are for the DALI ballasts and groups and will only appear if some DALI fixtures have been added to the project, see DALI. Note that each DALI interface has its own set of rows due to the restrictions of the DALI standard. Only DALI presets can be applied to this category.

AVC

These rows are for AVCs and will only appear if audio visual devices have been added to the project. Only AVC presets can be applied to these rows, see Working with the Audio Visual Controller.

Timeline row priorities

While the Latest Takes Precedent system determines what should be rendered and output as presets come and go over time, it is the order of the rows that determines what should be rendered and output should two or more presets *with fixtures in common* start simultaneously, with rows higher up the list taking precedent.

For example a fixture may be a member of two groups with a preset applied to both starting at the same time. In this case the fixture will render the preset for the group higher up the list. Groups can be reordered in the browser simply by dragging them to new positions.

Accordingly, simultaneous presets placed on groups and fixtures have a higher priority than presets placed on matrices.

Browser controls and feedback

The Browser toolbar provides controls for expanding and collapsing groups and compound fixtures as well as highlighting rows with programming. The Browser provides useful feedback as to which rows contain programming; elements, fixtures and groups will be displayed in blue, compound fixture or group headings will indicate the presence of any programming on members even when collapsed.

Expand all

Expands all compound fixtures and groups so that all element (pixel) rows are displayed. Items with programming will be shown in blue.

Expand all groups

Expands only groups so that all fixture rows are displayed. Items with any programming, even on a concealed member, will be shown in blue.

Collapse all

Collapses all so that only group rows are displayed. Groups with any programming, even on a concealed member, will be shown in blue.

Hide unused

Use this filter to hide all the unused rows, press again to turn off. Only items with any programming, even on a concealed member, will be shown in blue.

Selecting timelines

The timeline picker on the toolbar will be populated with every timeline in the project, select from the pull-down list to open a timeline for editing. Use Ctrl (Apple) + Tab to select the next timeline, Shift + Ctrl + Tab the previous.

Copying timelines

Timelines can be copied using the Copy Timeline button, the copy is a brand new instance that operates independently, useful for creating similar timelines.

Deleting timelines

Timelines can be deleted using the Delete Timeline button, a warning dialog will you prompt you to confirm. Note that there must be at least one timeline in the project.

Maintaining indefinite output

There are two ways of maintaining a timeline’s output beyond the end of the last preset. This is a particularly important feature for architectural use where a simple wall panel could be used to recall “scenes” at random which would remain active indefinitely until another is recalled:

⇨ Hold

Press the Hold button to prevent the timeline from releasing at the end (the default). Presets will remain active until overridden, effects and media will continue to play. Press the button again to reinstate the release.

Loop

Press the Loop Timeline button to make a timeline loop indefinitely. This is useful if you want to loop a sequence of presets. Press the button again to remove the loop.

It's also worth noting that a Timeline Running condition won't detect timelines that are holding at end. A Timeline On Stage condition will detect a looping or held at end timeline as long as the timeline is affecting the output of at least one fixture. Timeline Started and Timeline Ended triggers will match whenever a looping timeline loops. A Timeline Ended trigger will never match a timeline that is holding at end.

NOTE: Projects with lots of timelines set to Hold or Loop can eventually overwhelm the Controller(s) if these timelines are not explicitly released when no longer required.

Flags

Flags can be dropped onto timelines for use with triggers to create more complex presentations; perhaps incorporating remote sensors and conditional logic or triggering show control or AV equipment.

To set a flag, press the Add Trigger Flag button and drop it onto the timeline ruler at the required position. Hold down Ctrl (Apple) while pressing Add Trigger Flag to drop multiple in a single session, press the button again to finish.

To move a placed flag, drag it along the timeline ruler. To remove a placed flag, right-click on it and select Delete Flag.

Use the Triggers window to determine what these flags will do.

Learn timing

When simulating a single timeline, flags can be dropped interactively after pressing the Add Trigger Flag button to enter learn timing mode. Press F to drop a flag at each appropriate playback time then depress the Add Trigger Flag button to exit learn timing or click anywhere on the timeline (in which case a final flag will be dropped).

Locking timelines

When clicking on presets to edit their properties it is sometimes all too easy to move or stretch them by accident so it is possible to lock a timeline using the Lock button on the timeline toolbar. When a timeline is locked it is only possible to edit the preset properties, moving or stretching them is prohibited. Press the Lock button again to unlock.

Program - Working with presets

Applying presets

At its most basic let's, for example, make a fixture or group of fixtures go green. Select the Group folder and the "Fixed Colour" preset within and drag this onto the appropriate timeline row so that it starts at the required time, say at 0 seconds on the left. A 10 second long red strip (the default length and colour) will appear already selected for manipulation via the Preset Properties pane on the right. "Fixed Colour" preset properties are limited to colour and timing, use the colour picker to select green and set a fade time and skew as required.If you have the Simulate window open you can now simply click Start and you'll see these fixtures fade to green using the time and transition you have just entered and, after 10 seconds, fade back to black using the default release time of 2 seconds.

Presets can be moved and stretched on the timeline using the mouse to vary their start, end and length or alternatively you can type exact values into the Timing fields top right. Click View Transitions to display the fade and release timing graphically which can also be stretched using the mouse as an alternative to typing fade and release time values into the Timing pane.

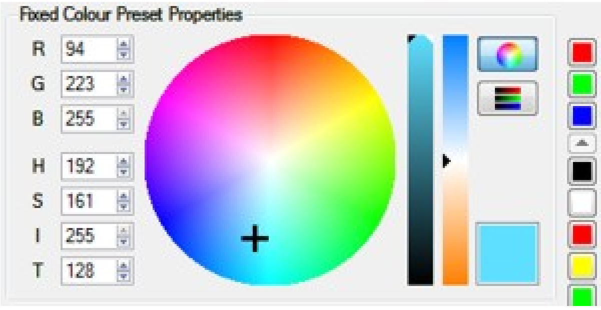
So getting slightly more adventurous let's say you want the fixtures to remain green for longer, say 20 seconds, and then fade to a rainbow effect. Firstly either drag the end of the green preset to 20 seconds on the timeline or set the end or length value to be 20 seconds via the Timing fields. Now select the "Rainbow Effect" preset (also in the Group folder) and drag this onto the timeline immediately following the green preset (so it starts at 20 seconds) and set its period to be 2 seconds with a "Spread" offset type. Again, use the Simulate window to view this new programming (click Reset then Start).

TIP! Hold Shift while dragging for finer resolution (centisecond). Hold Ctrl (Apple) while dragging to snap to the start/end of other placed presets.

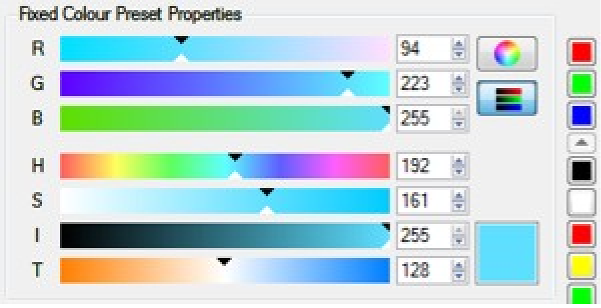
Programmed groups, fixtures or elements - i.e. those with at least one preset applied - are shown in blue in the Browser, unprogrammed remain black.

Colour picker & user palette

The majority of presets allow you to select one or more colours and so the colour picker and user palette is used to select a colour either graphically or numerically, two modes are provided:



R 94
G 223
B 255
H 192
S 161
I 255
T 128



R 94
G 223
B 255
H 192
S 161
I 255
T 128

A user colour palette is displayed on the right hand side. The top three buttons automatically keep track of the last three colours deployed and below them is the user palette which comes pre-loaded with useful primary and secondary colours, white and black. To add your own colour simply mix it using the picker, hold Ctrl (Apple) and click on a palette slot either empty or occupied, in the latter case the new colour will replace the old. If more slots are required then Ctrl (Apple) click on the plus sign at the bottom of the palette (you must first scroll down). The user palette is stored with the project.

Warm/Cold fixtures

The temperature slider takes effect on fixtures with warm white and cold white control channels and on fixtures with a single colour temperature channel.

Transparency

Some preset types support transparency. This is where one of the colours within the effect is specified as transparent, allowing whatever programming is running underneath to be seen. It is a very powerful feature allowing some very specific effects to be achieved that would otherwise be impossible.

However it also allows you to break some of the usual rules of Unison Mosaic playback and so it may need some extra thought or experimentation to get the result you are looking for. Here are some tips on how to get the best from this feature:

- Before using transparency make sure there isn't a way to achieve what you want using the existing opaque presets. If there is another way to get what you want then that may make life simpler. Use transparency for those very specific effects that cannot be achieved any other way.
- It can sometimes be difficult to make transparent effects behave tidily when a timeline loops. Use it in timelines that don't loop or hold at end when you can, or make a point of turning off the transparent effect before the point when it loops.
- It may be easier to get the result you want by using several timelines. Often problems can be avoided by having one timeline that contains the transparent effects and putting the background non-transparent effects into a separate timeline.
- When you are using multiple timelines, don't forget about the timeline priority setting. This can be a way of ensuring that transparent effects stay on top while you change the background underneath.

Mover presets

Mover presets can be dropped onto the Mover rows at the required time to synchronise their control with the rest of the presentation. Unlike the other presets, Mover presets remain active for the length of the timeline unless overridden by another Mover preset, such an override taking place by fixture parameter kind on the Latest Takes Precedent basis. For example, a Mover preset that only sets a fixture’s beam parameters will not interfere with a previous Mover preset that set its position. This position will remain active until such time that a Mover preset is encountered that does set this fixture's position in which case this new position will now become active.

TIP! It is possible to add more Mover rows (and delete them) so that more than two Mover presets can be run simultaneously, right-click>New Mover Row.

It is important to remember to set the intensity for movers by dragging the "Intensity" preset onto the timeline to determine when and for how long the movers are on. Movers that can colour mix should have a "Fixed Colour" or an effect preset applied to turn them on and set their colour.

DALI presets

DALI presets can be dropped onto the DALI ballasts and groups for each DALI interface in the project, see DALI. DALI presets should be thought of as commands instructing the DALI ballasts or groups to fade to a level or scene with the ballasts retaining this level or scene indefinitely regardless of the state of the timeline. Unlike DMX fixtures, there exists no notion of a released, default state and so DALI ballasts must be explicitly turned off with a preset. Beware that timelines set to loop will repeatedly run any placed DALI presets and thus reissue these commands until the timeline is released.

TIP! It may be simpler to separate DALI programming onto dedicated timelines and use triggers to synchronize them to the other fixture programming.

Copying presets

Presets can be copied (right-click>Copy) from one row and pasted (right-click>Paste) onto another (or later on the same row), this helps speed up the process of applying programming from one set of fixtures onto another; preset parameters, timing and transitions will all be copied. Note that copying presets in this way creates brand new instances of presets that operate independently of each other.

TIP! Hold Shift while selecting Paste to place the copy with finer resolution (centisecond).

Linking presets

If, however, you want to add more fixtures to an existing preset so as to operate on them all as one then drag the top or bottom edge of that preset up or down to include more rows of fixtures, this operation creates a linked preset. Note that any skewed timing or effects within the preset will now be rendered over the new, larger fixture selection - use Repeat to compensate if required.

A linked preset can be unlinked if desired by using right-click>Unlink to yield separate, identical instances.

Mover presets can not be linked since the fixture selection is a property of the preset itself.

Deleting presets

Presets can be deleted by pressing Delete,Backspace or using right-click>Delete.

Selecting multiple presets

To select more than one preset at a time for moving and editing properties or timing hold Ctrl (Apple) while clicking to build the selection or Ctrl (Apple) + A to select all.

Program - Working with the Audio Visual Controller

Overview

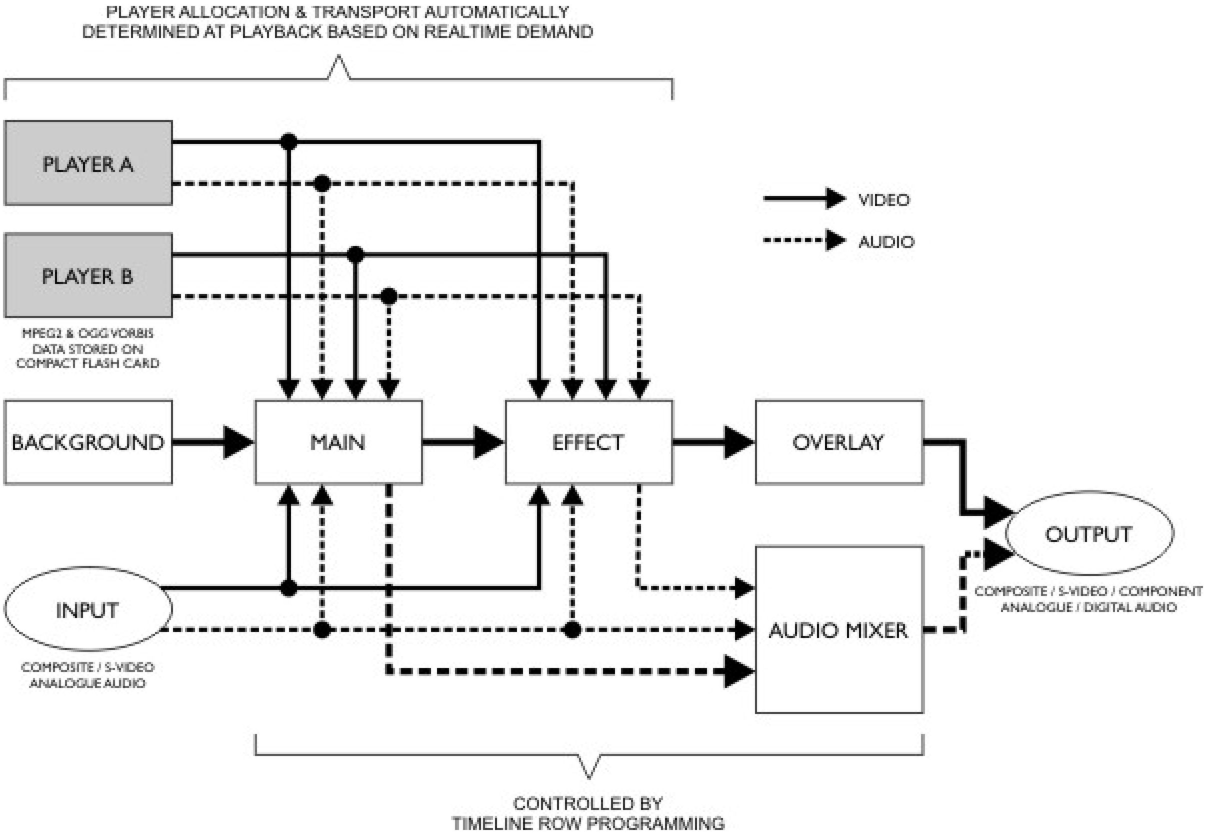
The Unison Mosaic Audio Visual Controller (AVC) is a companion to the Unison Mosaic MSCs. It allows the same level of intuitive control for audio and video that the MSCs provide for lighting.

Like the MSC, it is a rugged solid-state device in a compact DIN-rail mounting enclosure. Video and audio content is stored on a memory card (2GB card provided as standard) with instant access (no “seek” time) to any clip for playback. Video content is MPEG2 at full D1 resolution, that is DVD quality, and the aspect ratio can be either 4:3 or 16:9 anamorphic. Video output is selectable as composite, S-Video or component signals along with stereo balanced analogue.

However, the fundamental difference between the AVC and other video playback devices is that the AVC incorporates two separate players and a vision mixer. This means that it can respond immediately to external triggers and perform graceful transitions (dissolves, wipes, etc.) between two clips. This is a huge advantage for any sort of interactive or themed application and allows the compact AVC to replace multiple discrete items of larger AV equipment.

The AVC incorporates a live input so that a camera or other video feed can be mixed with its internal playback. There is also an internal effects unit supporting picture-in-picture and luma-key effects. Finally a separate overlay (downstream key) is supported which could either be a customer logo or text, including scrolling text from an external source.

Schematic of AV routing


PLAYER ALLOCATION & TRANSPORT AUTOMATICALLY DETERMINED AT PLAYBACK BASED ON REALTIME DEMAND
PLAYER A
PLAYER B
MPEG2 & OGG VORBIS DATA STORED ON COMPACT FLASH CARD
VIDEO
AUDIO
BACKGROUND
MAIN
EFFECT
OVERLAY
OUTPUT
COMPOSITE / S-VIDEO / COMPONENT ANALOGUE / DIGITAL AUDIO
AUDIO MIXER
INPUT
COMPOSITE / S-VIDEO ANALOGUE AUDIO
CONTROLLED BY TIMELINE ROW PROGRAMMING

Timeline Rows

Each AVC has four rows although only one is shown by default. Click the plus sign to expand this main row to expose these more advanced control rows. Only AVC presets can be applied to these rows, but not necessarily all AVC rows, and a preset's behaviour may change depending on the row on which it is placed:

Main

Used for the primary, full-screen image and/or audio.

Effect

Used for picture-in-picture (PIP) and luma-keyeffects:

- Fullscreen - use this if you want to play a brief clip over the top of that running on the main row and you do not want to stop and restart the main clip.
- Luma Key - will selectively superimpose parts of the effects clip based on the intensity (or luma) of the images. You can control the intensity threshold by setting the min and max values between 0 and 255. Over the range from min to max the two clips will be blended to make for a more seamless effect. You can then choose whether to superimpose the image from the effects clip directly or to superimpose a fixed colour of your choice.
- PIP - allows you to play the effects clip as a picture-in-picture. The picture-in-picture image will always be scaled down by half, but you can choose exactly where on screen to place the window using the X Position & Y Position settings. You can then choose whether to reduce the size of the window (perhaps to remove an unsightly border) and locate the image within the window using the X Offset & Y Offset settings. The "Correct aspect ratio" option is an advanced setting that can be used when the content being shown within the PIP does not match the aspect ratio of the screen - when enabled it will scale the content so that it looks visually correct.

Overlay

Used for dynamic text.

Audio Mixer

Used to override the audio levels. The default audio behaviour is to use the audio from the current clip playing on the main row, with the audio crossfading with the video. Here you can change the audio behaviour, for example to include audio from a clip playing on the effect row or from the live input, or simply to adjust volume levels.

AVC Presets

If there are no AV fixtures in the project, the AVC Presets folder will not be displayed.

Preset (user named)

Used to play the clips that you created using the Media window to import video and/or audio into your project. This preset requires a player and can be placed on either the main row (for full screen playback) or the effect row (for PIP or luma-key effects).

Block

Used in a multi-timeline playback scenario to temporarily (for the length of the block preset) inhibit row programming from other running timelines. This preset does not require a player and can be placed on any of the AVC rows.

Live input

Used to display live video and/or audio. This preset does not require a player and can be placed on the main row (for full screen playback) or the effect row (for PIP or luma-key effects).

Dynamic text

Renders a text message which can be changed at runtime:.

- Text Colour - the colour of the text
- Base Colour - the colour of the background
- Alpha - the transparency of the colour
- X Position/Y Position - the top left (in pixels) of the text's rectangle
- Width - the width (in pixels) of the text's rectangle (height is determined from the font)
- Period - the number of seconds that the message takes to scroll over the matrix
- Count - the number of times that the message should scroll over the length of the preset
- Text - the text to render
- Font - the font to use to render the text (see the Fonts dialog below)
- Refresh - when to update the text if it is changed (Immediately, On Loop, On Release)
- Scroll - select either static or the scroll direction of the text (Up, Down, Left, Right)
- Alignment - if the text is set to static, the alignment of the text (Left, Center, Right)
- Blend - if the text is set to scroll, specify a soft edge in pixels
- Seamless Loop - if the text is set to scroll, setting this will remove the gap between the end and the start
- Orientation - the orientation of the text (Horizontal, Vertical)
- Mirror - flips the text left to right
- Flip - flips the text top to bottom
- Effect - applies a dot matrix effect (None, Dot Matrix Square, Dot Matrix Circle)
- Dot Pitch - if a dot matrix effect is selected, sets the dot pitch in pixels of the pseudo matrix

NOTE: While this preset is complex, it may not achieve what you want, please contact support for assistance.

To configure the font used by the Dynamic Text preset, press the [...] button next to the font picker to open the Fonts dialog:

- Select a font from the Font picker
- Press New to create a new font
- Press Delete to delete the selected font (note that you cannot delete a font that is in use in the project)
- Set the font's name in the Name property
- Use Family, Size, Bold and Italic to set the appearance of the font
- Press Ok to close the Fonts dialog

NOTE: Editing a font will change all Dynamic text presets that use that font not just the currently selected preset (s).

The Dynamic Text preset allows you to change the text after uploading the project to a Controller. To do this, you need to specify which parts of the text are going to change and which parts will remain the same.

For example, to show the opening time of a venue, you might set the Text property to "Opening Time: <open>". This creates a text slot called 'open' which you can change the value of. You can have more than one slot specified in the Text property, for example "Opening Time: <open> Closing Time: <close>".

To set the initial text for a text slot, press the [...] button next to the Text property to open the Dynamic Text Configuration dialog:

- Click in the Default Value cell of a slot to edit the text stored in that slot
- You can remove unused text slots by pressing Remove
- Press Ok to save changes and Cancel to discard changes

There are two built-in slots, <time> and <date>, which show the current time and date respectively.

The Set Text Slot trigger action allows you to change the value of a text slot from a trigger.

Fixed colour

Renders a full screen fixed colour when placed on the main row. When this preset is used on the effects row, you can set an alpha value to blend the colour with the clip running on the main row. This preset does not require a player.

Audio mixer

Used to set the levels of the main, effect and live input audio, can only be placed on the audio mixer row. By default, only the main row audio is output but this preset allows you to alter the mix to include the effect and live input audio.

Timing & transitions

All the AVC presets provide control for an In and Out time, each with the option to select a transition effect (Dissolve, Wipe etc). Use the Transition pane with the preset selected on the timeline:

In

Specifies the duration of the transition, analogous to the fade time in lighting. The default is 0 seconds. When two clips abut on a timeline row it is the second clip's In time that is used to perform the overall transition from one to the other and not the first preset's Out time (see below).

Out

Specifies the duration of the fade out (release) of a preset when no preset immediately follows it on a timeline row. The default is 0 seconds.

Playback

Precedent

For flexibility and consistency with lighting playback each AVC row operates independently using the Latest Takes Precedent Plus (LTP+) system. It is thus possible,for example, to create timelines that contain just main row programming and timelines with just overlay row programming so that these can be mixed at playback allowing overlaid text to remain onscreen regardless of which video clip is chosen to play underneath.

Timeline row programming will remain onstage for the duration of the timeline unless overridden by another timeline with programming on the same row(s) or explicitly released.

Players & limitations

While the AVC's major advantage is that is has two separate players it is important to appreciate that it *only* has two players. The flexibility of both the programming interface and the Unison Mosaic system's multi-timeline environment can catch the unwary, particularly when using dissolves or wipes which require both players for the duration of such a transition.

Both players are routed to both the Main Mixer and the Effects unit (see schematic above) along with the Live input. The Play Clip preset requires exclusive use of a player and so only two Play Clip presets can run concurrently, either while transitioning from one to the other or with one main and the other as an effect (eg. picture-in-picture). Note however that a 0 second duration "snap" transition only requires a single player and the Live input feed none at all.

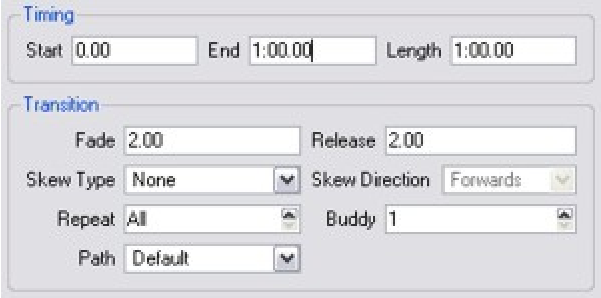
Which player does what is determined automatically at playback to ensure that multiple timelines interact correctly. Your programming does not specify a particular player, just the request to use one. When the programming on a single timeline, or the interaction of multiple timelines at playback, calls for more than the two players available the AVC will automatically replace any timed transitions with snap transitions so that it does not exceed the player limit.

Data storage

The AVC uses its memory card much more aggressively than the other Controllers and best performance requires the memory card be formatted in a particular way. It is therefore recommended that you format the memory card either from within Designer (Controller config) or the AVC's web interface when starting a fresh project or inserting a new card.

Timing, transitions & precedent

It is often said that good lighting as much about timing as anything else so it is important to understand the concepts of timing and transitions used throughout Designer:



Timing

Start 0.00 End 1:00.00 Length 1:00.00

Transition

Fade 2.00 Release 2.00

Skew Type None Skew Direction Forwards

Repeat All Buddy 1

Path Default

Timing

Timing values pertain to presets placed on timelines and determine the Start, End and Length times and may be numerically set as an alternative to dragging.

Transition timing

Transition values pertain to Mover preset defaults and presets placed on timelines and determine what sort of cross fade is rendered. The use of interesting transitions can transform your project so it is well worth spending some time experimenting to see what can be achieved using these properties:

Fade

Sets the overall length of the transition, default value is 2 seconds.

Release

Sets the time used for a preset to “release” its fixtures when it completes, default value is 2 seconds.

Path

Specifies the cross fade path, default is Default meaning that each pixel or parameter will use the library default path (or Dimmer Curve if specified), typically Linear for intensity/position and Start for colour/gobo wheels. Some interesting additional paths are provided for your delectation.

Transition skews - Group & Mover presets (1D)

Skew Type

Various different skews can be selected to alter how individual fixtures (and pixels within fixtures in the case of compound fixtures such as battens and tiles) behave within the fade, default is None which means that all fixtures/pixels fade together. Use skews to create “multi-part” fades so that fixtures/pixels fly in one by one for example (set to Individually) - you may have to increase the fade time to clearly perceive the skew especially with lots of fixtures/pixels.

Direction

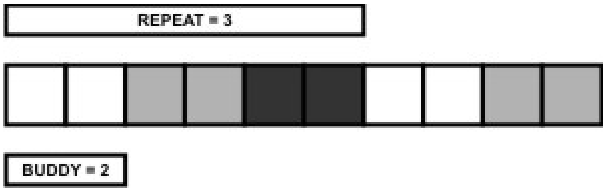
The ordering of a skewed transition depends on the fixture/pixel ordering within the group. The Skew Direction drop-down provides further ordering options such as Forwards and Backwards for additional flexibility. Additional groups can be created with different fixture/pixel ordering to achieve other skewed effects.

Repeat

Specifies the number of adjacent fixtures/pixels over which a skewed transition is applied, default is All meaning that the skew will span the entire selection. Typically you set this value to be equal to the number of pixels in a compound fixture or the number of fixtures in a zone or on a truss, experimentation is recommended as interesting effects can be achieved.

Buddy

Specifies the number of fixtures/pixels that will fade together within a skewed transition, default is 1 meaning that each fixture/pixel will fade independently. Set to 2 to make pairs fade together, 3 for threesomes etc. Again, experimentation recommended.



Transition skews - Matrix & Media presets (2D)

Skew Type

Since the Pixel Matrices onto which you place Matrix and Media presets have spatial awareness, the available skews are more powerful and are akin to video wipes - you may have to increase the fade time to clearly perceive the skew.

% Fade

Sets the hardness of the skew; whether the edge of the wipe is hard (0%) or soft (100%).

Angle

Some skews (for example Linear or Radial Wipe) can optionally accept an angle value that alters the direction or start point of the transition.

Specifying times

Timing fields display times in the format hh:mm:ss.cc (hours:minutes:seconds.centiseconds), although leading zeros are not displayed. 24 hours (24:00:00.00) is the maximum timeline length and thus timing duration.

When setting times you can enter in this format directly (omitting leading zeros) or you can use h, m & s to specify your units and Designer will reformat accordingly. Furthermore, any number input without separators (h, m, s or :) is taken literally if it is valid as such or converted if not, here a decimal point will always denote centiseconds.

For example:

00:01:30.00	1 minute and 30 seconds	(00:01:30.00)
1:30	1 minute and 30 seconds	(00:01:30.00)
90s	1 minute and 30 seconds	(00:01:30.00)
1h2.5m	1 hour, 2 minutes and 30 seconds	(01:02:30.00)
2h7m45.5s	2 hours, 7 minutes, 45 seconds and 50 centiseconds (half a second)	(02:07:45.50)
99	1 minute, 39 seconds ("99" not valid so converted)	(00:01:39.00)
100	1 minute,0 seconds ("100" valid so taken literally)	(00:01:00.00)
2020	20 minutes, 20 second	(00:20:20.00)
30.1	30 seconds and 10 centiseconds (tenth of a second)	(00:00:30.10)

Precedent

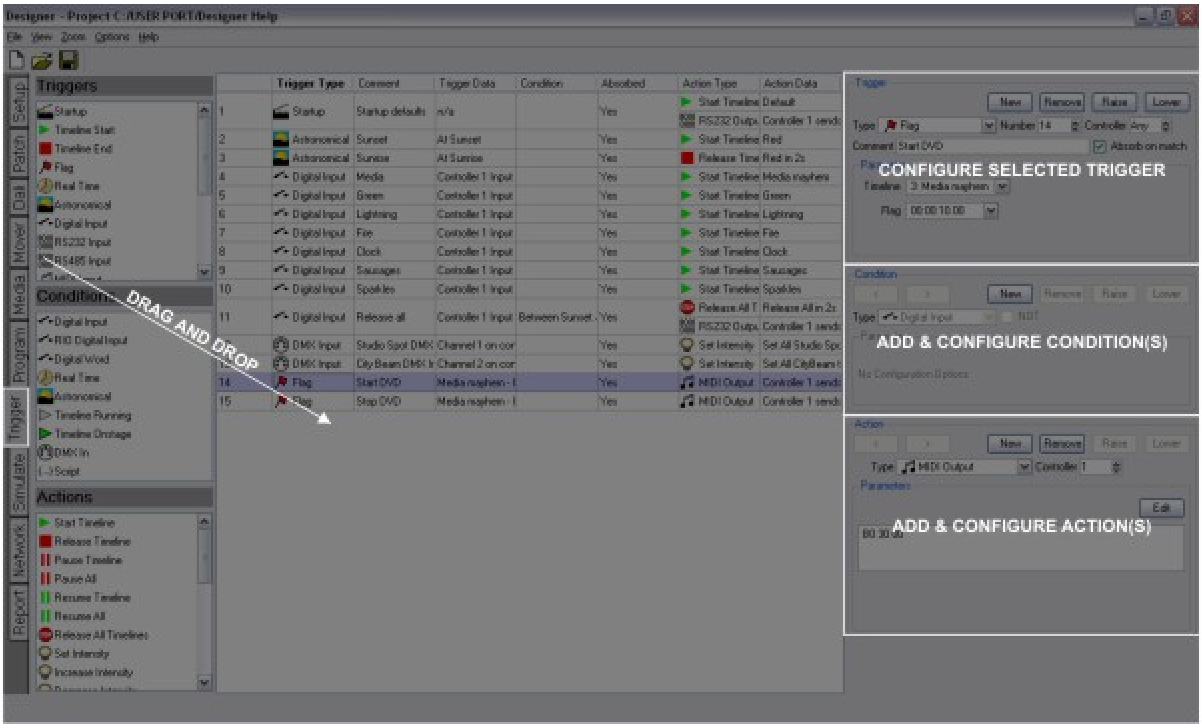
The Unison Mosaic Controllers use the Latest Takes Precedent Plus (LTP+) system (popularised by Flying Pig Systems in the early 1990s) to determine what to output to a fixture (or, strictly speaking, fixture element or parameter) at playback runtime. LTP+ was an enhancement of the standard LTP system and was designed to incorporate automated lighting control. The "rules" of the LTP+ system are as follows:

1. After system initialisation, and prior to any preset (on a timeline) running, the output will be in a default, "released" state. This does not mean that all DMX channels will be zero however as this default state is determined by each fixture's library definition that will set parameters to sensible "home" positions, for example pan and tilt to midway, irises and gobo/colour flags to open white.
2. The output will respond to the latest preset activated regardless of the preset data (so a preset programmed to black will override a colour).
3. When a preset expires or is explicitly released the output will revert to the prior state which may be an overridden preset (if any) or the default state.
4. Fixture parameters are grouped by kind (Intensity, Colour, Beam Image, Beam Shape, Position & Control) and so preset programming and thus precedent operates at this level - multiple presets can thus be responsible for the output of a fixture with multiple parameter kinds e.g. a moving light or conventional fixture with a scroller.

Since the Controllers can run multiple timelines simultaneously then some consideration should be given as to how best structure the project. This is particularly important if the project calls for random triggering of timelines, for example from a Button Panel Station (BPS), since there is no way of knowing in advance what state the output will be in (i.e. which timelines have already been triggered) when a new timeline triggered. This interaction of timelines can yield unexpected results unless care is taken when programming.

Trigger - Overview

The Trigger window is used to “connect” your timeline programming to the outside world:



Designer - Project C:/USER PORT/Designer Help

	Trigger Type	Comment	Trigger Data	Condition	Absorbed	Action Type	Action Data
1	Startup	Startup defaults	n/a		Yes	Start Timeline RS232 Outpu	Default Controller 1 send
2	Astronomical	Sunset	At Sunset		Yes	Start Timeline	Red
3	Astronomical	Sunrise	At Sunrise		Yes	Release Time	Red in 2s
4	Digital Input	Media	Controller 1 Input		Yes	Start Timeline	Media mayhem
5	Digital Input	Green	Controller 1 Input		Yes	Start Timeline	Green
6	Digital Input	Lightning	Controller 1 Input		Yes	Start Timeline	Lightning
7	Digital Input	Fire	Controller 1 Input		Yes	Start Timeline	Fire
8	Digital Input	Clock	Controller 1 Input		Yes	Start Timeline	Clock
9	Digital Input	Sausages	Controller 1 Input		Yes	Start Timeline	Sausages
10	Digital Input	Sparkles	Controller 1 Input		Yes	Start Timeline	Sparkles
11	Digital Input	Release all	Controller 1 Input	Between Sunset	Yes	Release All T RS232 Outpu	Release All in 2s Controller 1 send
12	DMX Input	Studio Spot DMX	Channel 1 on cor		Yes	Set Intensity	Set All Studio Spo
13	DMX Input	City Beam DMX I	Channel 2 on cor		Yes	Set Intensity	Set All CityBeam
14	Flag	Start DVD	Media mayhem - 1		Yes	MIDI Output	Controller 1 send
15	Flag	Stop DVD	Media mayhem - 1		Yes	MIDI Output	Controller 1 send

Triggers
Condition
Actions
DRAG AND DROP
CONFIGURE SELECTED TRIGGER
ADD & CONFIGURE CONDITION(S)
ADD & CONFIGURE ACTION(S)

Controllers support a range of interfaces which can be used to trigger the playback engine including an internal realtime & astronomical clock. For example, digital input #1 (connected to a wall panel) could be set to start “Funky” timeline, “Advert” can be set to run on the hour every hour between sunrise and sunset and, at sunset, “Cleaning” would start.

The window comprises three sections: On the left are the Trigger, Condition and Action directories. In the middle is the trigger management area and the panes to configure the Triggers, Conditions and Actions are on the right.

Creating a trigger

To create a trigger, select and drag it from the directory and drop it onto the management area or press the ✚ button on the Trigger configuration pane and use the pull-down menu to select the required type, see basic, advanced and Remote Device triggers.

Configuring a trigger

- Type - the trigger type
- Number - the trigger's unique number as used for web interface control purposes (can typically be left at the default value unless creating custom pages)
- Controller - the controller that will process the trigger, note that Realtime, Astronomical and Ethernet Input triggers are processed by the Time Server so the selection is limited to None or Time
- Comment - enter a useful name for simulation and web interface control purposes
- Absorb on match - uncheck to prevent the trigger absorbing the match, see below
- Parameters - the data required for each trigger type, varies by type so refer to the appropriate trigger descriptions

Inhibiting a trigger

For testing purposes it is sometimes useful to inhibit one or more triggers to examine more clearly the operation of others. A trigger can be inhibited by setting the controller number to None so that no controller processes the trigger, the row details will be displayed in grey.

Incompletely defined triggers

If triggers/conditions/actions are not completely defined then they are highlighted in red to indicate that they need further attention.

Trigger order & matching

The order in which triggers are displayed in the Trigger screen is the order in which they are tested by the system. Once a trigger is successfully matched then,if "Absorb on match"is checked, no further triggers are tested for that event; the event is absorbed.Thus this trigger order is important, particularly when using conditions.

If you had two identical triggers in your show then, assuming they had no conditions, only the first one encountered would ever be matched. However, if you add a condition to the first trigger then it will only match when the condition is true, and when it is false the second trigger will match instead.

The ability to have the same trigger have different results based on a condition is very powerful. For instance you might have a single digital input that starts one timeline during the day and another during the night.

Changing the trigger order

You can either select and drag the trigger up or down within the management area to redefine the order or use the ⬆ and ⬇ buttons on the Trigger configuration pane.

Absorb on match

In some cases it is useful for a matched trigger not to absorb the event and thus allow triggers further down the list, so there exists the option to disable the default behaviour (for triggers other than Realtime & Astronomical clock which default to unchecked) by unchecking the "Absorb on match" box for each trigger as required.

Conditions

If you wish to constrain the trigger with a condition then drag one from the directory and drop it into the Condition field of the trigger or press the ✚ button on the Condition configuration pane and use the pull-down menu to select the required type, see conditions.

Up to 32 conditions can be applied to each trigger in this way and you can use the < and > buttons to select each one for configuration.

Configuring a condition

- Type - the condition type
- NOT - check to invert the operation of the condition
- Parameters - the data required for each condition type, varies by type so refer to the appropriate condition descriptions

Removing a condition

Use the ⇦ and ⇨ buttons to select the condition to be removed and press the − button on the Condition configuration pane.

Changing the condition order

To change the order in which conditions are tested, use the ⇧ and ⇩ buttons on the Condition configuration pane.

Actions

Every trigger needs an action, the thing to do, which you can drag from the directory onto the Action field of the trigger or press the + button on the Action configuration pane and use the pull-down menu to select the required type, see basic, advanced and Remote Device actions.

Up to 32 actions can be applied to each trigger in this way and you can use the ⇦ and ⇨ buttons to select each one for configuration.

Configuring an action

- Type - the action type
- Controller - the controller that will process the action
- Parameters - the data required for each action type, varies by type so refer to the appropriate action descriptions

Removing an action

Use the ⇦ and ⇨ buttons to select the action to be removed and press the − button on the Action configuration pane.

Changing the action order

To change the order in which actions are executed, use the ⇧ and ⇩ buttons on the Action configuration pane.

Copying a trigger

Select the trigger to be copied in the management area and use right-click > Copy to create a copy in the row immediately below, very useful for defining similar triggers since all the defined conditions and actions are also copied.

Deleting a trigger

Select the trigger in the management area and use right-click > Delete or press the − button on the Trigger configuration pane, the entire trigger, its conditions and actions will be deleted.

Triggers - Basic

These basic triggers allow you to use the Controller's realtime and astronomical clock, digital inputs and web interface to control timeline playback and set intensity levels:

Basic trigger types

Startup

The startup trigger determines what the Controller should do after power up or reset. There are no configuration options.

Timeline Started

A timeline starting (generally as a result of a trigger or the timeline looping) can be used as a trigger, use the configuration pane to select which timeline.

Timeline Ended

A timeline reaching the end of its programming can be used as a trigger, use the configuration pane to select which timeline. For a looping timeline, this trigger will fire every time the timeline loops.

Timeline Released

A timeline releasing can be used as a trigger, use the configuration pane to select which timeline.

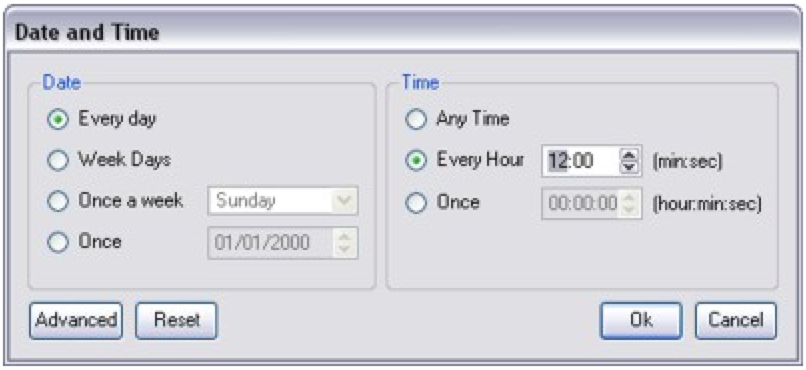
Timeline Flag

Any timeline can have one or more flags placed on the ruler (see working with timelines) to act as triggers, use the configuration pane to select which timeline and the flag within that timeline.

Real Time

The Controller has an internal, battery-backed real time clock. In a project with multiple Controllers only one Controller is set as the Time Server (see Controller association), use the configuration settings to determine what sort of real time event will be matched, for example 5 minutes past every hour or at noon on a specific date.

The standard dialog allows you to deal with the most common cases, including one-off events or events that recur hourly, daily or weekly. Note that the maximum resolution of real time events is 1 second, so an "Any Time" trigger will fire every second during the specified date range:

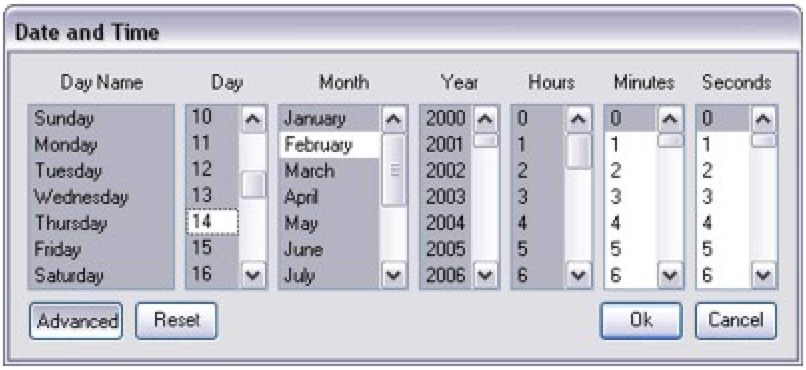


Date and Time

Date
● Every day
○ Week Days
○ Once a week Sunday
○ Once 01/01/2000

Time
○ Any Time
● Every Hour 12:00 (min:sec)
○ Once 00:00:00 (hour:min:sec)

There is also an advanced dialog that allows you to specify a precise mask of when the trigger should fire, using a combination of year, month, day of the month, day of the week, hour, minute or second. Highlighted values are included in the mask and make sure all values are highlighted in any column you don't care about. The trigger will fire at all times that match the specified mask in all columns - so no column should be blank or the trigger will never match:



Date and Time

Day Name	Day	Month	Year	Hours	Minutes	Seconds
Sunday	10	January	2000	0	0	0
Monday	11	February	2001	1	1	1
Tuesday	12	March	2002	2	2	2
Wednesday	13	April	2003	3	3	3
Thursday	14	May	2004	4	4	4
Friday	15	June	2005	5	5	5
Saturday	16	July	2006	6	6	6

Further information about the use of the real time clock can be found in the conditions chapter.

In the Network view, a Controller properties option exists to "execute real time triggers on startup". This will ensure that all real time triggers are executed from a user-specified time to the current time to reinstate the correct playback state in case the Controller is restarted for some reason (eg.power loss, watchdog or remote reset).

NOTE: Real Time triggers are only tested by the Time Server and then shared over the network so any conditions are tested on the Time Server only.

Astronomical

The Controller is also equipped with astronomical clock algorithms which automatically generate the correct sunrise, sunset and twilight times for the location of the installation (see project properties), use the configuration pane to select between sunrise, sunset, dawn or dusk and to specify an offset, negative or positive, in minutes.

Two versions of dawn and dusk are offered, using the two definitions of twilight: *civil* and *nautical*. Please see Wikipedia for an explanation of these terms.

A Controller properties option exists to ensure that all astronomical triggers are executed from a user-specified time to the current time to reinstate the correct playback state in case the Controller is restarted for some reason (eg.power loss, watchdog or remote reset).

NOTE: Astronomical triggers are only tested by the Time Server and then shared over the network so any conditions are tested on the Time Server only.

Lunar

As well as astronomical triggers the Controller uses lunar clock algorithms to calculate the lunar phases based on the location of the Controller (see project properties).

The lunar events are new moon, first quarter, full moon and third quarter. Use the configuration pane on the right to select the phase.

NOTE: Lunar triggers are only tested by the Time Server and then shared over the network so any conditions are tested on the Time Server only.

Digital Input

The MSC and TPC with TPC-RIO has 8 digital inputs which can be used as triggers, either to detect a voltage or a contact closure. Use the configuration pane to select which Controller (None, Any or a particular MSC or TPC with TPC-RIO), which Input (1 thru 8) and the polarity of the logic - select Low for contact closure or when driving with an "active low" signal, select High for driving with an "active high" signal.

To receive a digital input from a Unison Mosaic Remote Device, change the Device to RIO 80 or RIO 44 and set the RIO number, or leave as *Any*. The RIO 80 has 8 inputs and the RIO 44 has 4 inputs.

The inputs on the revised MSC and TPC with TPC-RIO hardware and the RIOs can also be configured as analog inputs - see Advanced Triggers.

Soft Trigger

This trigger type is provided for triggering from the web interface, there are no configuration options.

Basic action types

Start Timeline

Starts a timeline, use the configuration pane to select which timeline.

Release Timeline

Releases a timeline, use the configuration pane to select which timeline.

Toggle Timeline

Starts a timeline if it's not running, or releases the timeline if it is.

Pause Timeline

Pauses a timeline at its current position - effects and media will also freeze, use the configuration pane to select which timeline.

Pause All

Pauses all timelines at their current position - effects and media will also freeze.

Resume Timeline

Resumes playback of a paused timeline from its current position, use the configuration pane to select which timeline.

Resume All

Resumes playback of all paused timelines from their current positions.

Release All Timelines

Releases all output, use the configuration pane to set a release fade time.

Set Intensity

Sets the intensity of a group of fixtures, use the configuration pane to select the group, intensity level, fade and delay times. Because the MSCs are genuine lighting controllers as opposed to DMX framestore devices, realtime control of intensity is available at all times as it would be on a sophisticated lighting console. You can control the intensity of one or more groups of fixtures regardless of what timeline(s) they may be running.

You can think of each group as having its own intensity fader, which this action allows you to move between 100% (default) and 0%. You can specify which group to affect and the new position for the fader. It is sometimes useful to set the fader position (as a percentage) from a variable - this permits direct intensity mastering via an input such as serial, MIDI or DMX. The fade and delay times can also be set from variables.

The fader modifies the programmed intensity for all fixtures within the group. On startup all groups have their faders at 100%. Where multiple groups containing the same fixtures have their intensity reduced then the decrease is cumulative.

Note that if you decrease intensity for one group you can only increase it again by acting on the same group. Applying an increase intensity action to a different group will have no effect even if that group contains the same fixtures - you would be trying to move a different fader.

Increase & Decrease Intensity

Increases or decreases the intensity of a group of fixtures, use the configuration pane to select the, group, step size (in percent), fade and delay times. Because the MSCs are genuine lighting controllers as opposed to DMX framestore devices, realtime control of intensity is available at all times as it would be on a sophisticated lighting console. You can control the intensity of one or more groups of fixtures regardless of what timeline(s) they may be running.

You can think of each group as having its own intensity fader, which these actions allow you to move between 100% (default) and 0%. You can specify which group to affect and the increment by which to change the fader position. It is sometimes useful to set the step size, fade and delay times from variables.

The fader modifies the programmed intensity for all fixtures within the group. On startup all groups have their faders at 100%. Where multiple groups containing the same fixtures have their intensity reduced then the decrease is cumulative.

Note that if you decrease intensity for one group you can only increase it again by acting on the same group. Applying an increase intensity action to a different group will have no effect even if that group contains the same fixtures - you would be trying to move a different fader.

Set AVC Master

Sets the video brightness and audio volume of the output of an AVC.

Triggers - Advanced

Unison Mosaic Controllers support numerous serial communication ports and optional Remote Devices and the following advanced triggers & actions, together with conditions, variables and Lua scripts, provide powerful show control functionality. Indeed, above and beyond a Controller's lighting & AV control capabilities, a Controller can be used to tie together numerous pieces of equipment with various interfaces to yield a robust and fully integrated show control solution.

Advanced trigger types

Analog Input

The revised MSC and TPC with TPC-RIO hardware has 8 inputs that can be configured as digital or analog inputs in the Interfaces tab of the Network view. The RIO 80 and RIO 44 have inputs that can be configured as digital or analog inputs in the Remote Devices tab of the Network view.

Use the Controller and Input settings to specify which Controller’s analog input should be considered the input source. Alternatively, leave the Input set to *Any* to match any of the inputs of the Controller and to capture the input as a variable. To use a RIO's input as the input source, change the Device from Local and select the RIO number, or leave this as *Any*.

Now you should specify the range of voltage to trigger on. You can choose whether to trigger every time the voltage changes within the specified range ("Changes in range"), or to only trigger when the voltage enters the specified range ("Enters range"). "Enters range" is generally more useful when you are using analog inputs to trigger timelines, but "Changes in range" would be required if you were using an analog input as a variable passed to a Set Intensity action to control the intensity for a group.

The voltage range of a Controller's or RIO's analog input can be configured in the Network view. The smallest measurable voltage change is 0.25V.

Serial Input

RS232 remains a very popular protocol for interfacing equipment and the RS232 port of a Controller or Remote Device can be configured to support most common data formats. RS485 is a more robust alternative to RS232 (better noise immunity, longer cable lengths and faster data rates) and is a widely supported protocol. A Controller or Remote Device can be configured to receive RS232 full-duplex or RS485 half-duplex in the Network view, see Controller interfaces and Remote Devices. A TPC with TPC-RIO can receive RS232 full-duplex.

To receive serial from a Controller's serial port, leave Device as Local and use the Controller setting to specify which Controller's serial port should be considered the input source.

If you're using an RS485 expansion module with the original MSC hardware, select which module should be the input source by setting the number next to the Device:

1	Controller serial port
2	1st RS485 expansion module
3	2nd RS485 expansion module (only used if two RS485 modules are connected)

For the MSC X, the number next to the Device setting selects which of the two RS232 ports should be the input source.

Alternatively, set the Device to a RIO and select the RIO number.

Now define the string of input characters to be matched as the trigger. There are three formats in which serial strings can be entered:

Hex	A series of hexadecimal characters (0-9, a-f, A-F) where pairs of values are interpreted as a byte.
Decimal	A series of decimal characters (0-255) separated by "." characters.
ASCII	A series of ASCII characters. The special characters '\n' for new line, '\r' for carriage return, and '\t' for tab are supported.

Additionally, each byte can be replaced with a wildcard to match a range of input characters and these wildcards can even be captured as variables to determine the trigger's action.

Ethernet Input

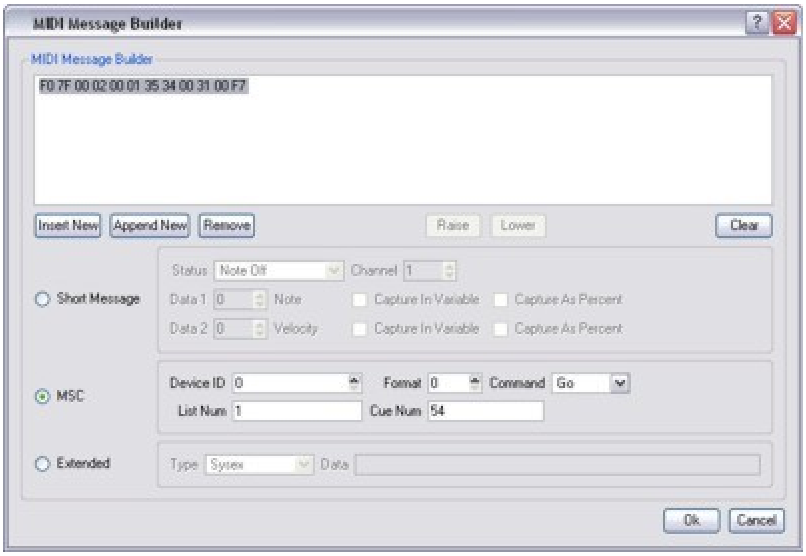
Use the Controller setting to specify which Controller should process the Ethernet input. Select the Ethernet Source (see Controller interfaces) and press Edit to define the string of input characters to be matched as the trigger in much the same way as RS232 (see above).

MIDI Input

MIDI is another very popular protocol for interfacing equipment and the MIDI input trigger allows you to define, via a convenient MIDI Message Builder, the type (Short message, MIDI Show Control or Extended) and command string that is to be matched as the trigger. Variables can be captured to determine the trigger's action.

Use the Controller setting to specify which MSC's MIDI port should be considered the input source. To use the MIDI port on a RIO A, set the Device to RIO A and specify the RIO A number, or leave this as *Any*. In this case, the RIO number will be captured as a variable.

Press Edit to open the Message Builder:



MIDI Message Builder

MIDI Message Builder

F0 7F 00 02 00 01 35 34 00 31 00 F7

○ Short Message

Status Note Off Channel 1

Data 1 0 Note □ Capture In Variable □ Capture As Percent

Data 2 0 Velocity □ Capture In Variable □ Capture As Percent

● MSC

Device ID 0 Format 0 Command Go

List Num 1 Cue Num 54

○ Extended

Type Sysex Data

Press Insert New, select one of the three message types and then the specific command and variables.

Press Append New to add and define another command string to be matched, the Raise and Lower buttons can be used to define the matching order.

Press Remove to delete a command string and Clear to delete them all.

The resulting hexadecimal string will be constructed automatically and displayed in the window for reference with question marks ("??") indicating undefined characters in MIDI Show Control (since we do not know in advance how many characters will be captured) or <c>, <d> and <x> as appropriate for Short and Extended messages.

Press Ok to finish.

Use Options > Preferences > Triggers to select the default Short MIDI message data format.

A comprehensive guide to MIDI is beyond the scope of this document, see the MIDI Manufacturers Association for more details, and the manual for the equipment to be interfaced will also certainly be an invaluable reference.

DMX Input

DMX-IN Expansion Module required for original MSC hardware. Revised MSC hardware, AVC and MSC X can receive DMX directly. TPC and MSC X can receive DMX-In via Art-Net and sACN.

Use the Controller setting to specify which controller should receive the DMX. If using two DMX-IN Expansion Modules with an MSC, use the Port setting to specify which module should be considered the input source (modules are numbered in the order they are connected to the MSC).

Now you should specify which DMX channel to look at and the range of values to trigger on. You can choose whether to trigger every time the value changes within the specified range ("Changes in range"), or to only trigger when the value enters the specified range ("Enters range"). "Enters range" is generally more useful when you are using DMX to trigger timelines, but "Changes in range" would be required if you were using a DMX channel as a variable passed to a Set Intensity action to control the intensity for a group.

Audio Input

The RIO A has a stereo balanced line level audio input that can be used as a trigger. The original MSC hardware can be used with an audio expansion module, though the RIO A offers a great advance in performance.

To use an MSC with an audio expansion module as the input source, use the Controller setting to specify which Controller has the expansion module and leave the Device as Local. If two audio modules are connected, set the port number to match which module should be the input source.

To trigger from a RIO A, set Device to RIO A and select the number of the RIO A, or leave this set to *Any* to cause the trigger to attempt to match against audio input from any RIO A. In this case, the RIO number will be captured as variable.

Use the Channel setting to specify whether the trigger should match against the left or right audio channel, or the combination of the two (only available for RIO A). Now select which frequency band to use, or leave this set to the overall volume of the channel. Each audio module analyses the incoming audio as 5 frequency bands. Each RIO A can analyse incoming audio as up to 30 frequency bands - see Remote Devices.

The Peak checkbox tells the trigger to match on the decaying level of the last peak in the audio frequency band.

Finally, specify the range of values to trigger on. You can choose whether to trigger every time the value changes within the specified range ("Changes in range"), or to only trigger when the value enters the specified range ("Enters range"). "Enters range" is generally more useful when you are using audio to trigger timelines, but "Changes in range" would be required if you were using an audio band as a variable passed to a Set Intensity action to control the intensity for a group.

DALI Ballast Error

RIO D, TPC with TPC-RIO or DALI-S Expansion Module required.

Use this to trigger from a DALI ballast reporting an error. Specify the interface then use All to match if any ballast on that interface reports an error. Alternatively select a single address to match to. Next select the error type to match to.

DALI Input

RIO D, TPC with TPC-RIO or DALI-S Expansion Module required.

To use an MSC with a DALI-S expansion module as the input source, use the Controller setting to specify which Controller has the expansion module and leave the Device as Local. If two DALI-S modules are connected, set the port number to match which module should be the input source.

To use a TPC with TPC-RIO as the input source, use the Controller setting to specify which Controller has the TPC-RIO and leave the Device as Local.

To trigger from a RIO D, set Device to RIO D and select the number of the RIO D, or leave this set to *Any* to cause the trigger to attempt to match against DALI input from any RIO D. In this case, the RIO D number will be captured as variable.

The RIO D, TPC with TPC-RIO and DALI-S Expansion Module snoop the DALI bus and so the trigger can be set up to respond to any DALI commands:

- Command - select Direct Level (0>254), Scene or Relative Level
- Address - select All, Group (0>15) or Ballast (1>64)
- Min/Max - select the level to match for Direct Level triggering or
- Scene - select the scene (1>16) for Scene matching or
- Type - select the type of Relative Level command to match

The RIO D and TPC-RIO both recognise DALI input from Light Sensors and Occupancy Sensors that utilise eDALI. When triggering from an Occupancy Sensor select which state is to be matched. When using a Light Sensor, specify what range of light level (0>254) is to be matched. See the table below for light levels:

Light Sensor Level	**Lux Range**
0 - 31	0.00 - 7.75
32 - 63	8.00 - 15.75
64 - 95	16.00 - 31.75
96 - 127	32.00 - 63.75
128 - 159	64.00 - 127.75
160 - 191	128.00 - 255.75
192 - 223	256.00 - 511.75
224 - 254	512.00 - 1008.00

DALI Bus Power

RIO D or TPC with TPC-RIO required.

Use this trigger if you want to act upon a change of the electrical state of a specific DALI bus. Buses can be in one of three states: Correct Power, Incorrect Power and No Power.

BPS Button

The BPS has eight buttons which can be used as triggers.

Use the configuration pane to select which Controller should process the trigger. Select the BPS, button number (or leave as *Any* - see variables) and the type of button event (Press, Held, Repeat, Release). Setting the button number to *Any* will capture the pressed button as a variable.

Paradigm Indicator

Use this trigger for integration with a Unison Paradigm system. Specify an Indicator Type and then the relevant trigger matching information for that indicator. Setting a parameter to *Any* will capture the value as a variable.

Note: To use this trigger ensure a Paradigm Processor IP address has been specified in the Network tab.

Remote Device Online

Use this trigger, not the Startup trigger (which will fire before the Remote Devices can be detected), if you wish to act upon the detection of a Remote Device, for example to configure it with settings other than its defaults.

Use the configuration pane to select which Controller should process the trigger and select the Remote Device's type and identification number (or leave as *Any* - see variables).

Remote Device Offline

Use this trigger if you wish to act upon the loss of a Remote Device, for example to enter a failsafe state and issue a warning.

Use the configuration pane to select which Controller should process the trigger and select the Remote Device's type and number (or leave as *Any* - see variables).

Advanced action types

Set Timeline Rate

The set timeline rate action allows the playback speed of a particular timeline to be modified on the fly. You can select which timeline you want to control or get the timeline number from a variable. The rate is specified as a percentage, where 100% is the programmed rate, 200% would be double speed, and 50% would be half speed. The rate can also be driven by a variable.

Set Timeline Position

The set timeline position action allows the playback position of a particular timeline to be modified on the fly, typically the timeline would be paused to prevent it running on by itself. You can select which timeline you want to control or get the timeline number from a variable. The position is specified as a percentage, where 0% is the start and 100% the end. The position can also be driven by a variable.

Set Timecode

Use this to set the timecode position for one of the six Time Sources (set the appropriate timecode format).

Set Fixture RGB

Use this to inhibit a fixture's RGB levels selectively either to a fixed value or to track a variable. The latter makes for some very interesting realtime effects when used in conjunction with 2D and Media presets. Set a fade time to introduce the change.

Clear Fixture RGB

Use this to clear one or all fixture RGB inhibitions (see above), set a fade time to release the change(s).

Set DALI Level

Use this to set an intensity level on DALI fixtures patched to a specified Interface. You can specify whether you want all of the fixtures on that Interface, fixtures in a certain group or a single fixture to be affected by the level change. You can also specify a fade time or choose to reuse the last fade time stored on the ballasts.

Recall DALI Scene

Use this to a DALI scene on DALI fixtures patched to a specified Interface. You can specify whether you want all of the fixtures on that Interface, fixtures in a certain group or a single fixture to be affected by the scene change. You can also specify a fade time or choose to reuse the last fade time stored on the ballasts.

DALI Command

Use this to send a DALI command to DALI fixtures patched to a specified Interface. You can specify whether you want all of the fixtures on that Interface, fixtures in a certain group or a single fixture to be affected by the command. DALI commands include off, fade up/down, step up/down, step to min/max, step down and off and step up and on. Where fading is involved, you can specify a fade rate or choose to reuse the last fade rate stored on the ballasts.

Mark DALI Ballast Fixed

Use this to mark all or a single emergency ballast as fixed on the specified interface.

DALI Start Emergency Test

Use this to start a Duration or Function test on all or a single DALI address(es) on the specified interface.

DALI Stop Emergency Test

Use this to stop a Duration or Function test on all or a single DALI address(es) on the specified interface.

Set Text Slot

The Set Text Slot trigger action allows you to change the value of a text slot from a trigger, see the Dynamic Text preset.

You select the slot either by picking from the Text Slot list or by specifying a variable. If you use a variable, the variable must have captured a string in the trigger, and that string must be the name of an existing text slot.

The value to put in the text slot is then selected with the second variable.

Inject Trigger

Use this to fire a Trigger. You can specify which trigger you want to fire or which variable you want to use to get the trigger number from. You can also choose whether you want test the conditions of that trigger or not.

Run Script

Use this to run a Lua script, press Launch Editor to open the script editing dialog. If you can not achieve what you want with the triggers and actions provided it is almost certain that a script can be defined to solve your problem.

Unison Mosaic Controllers support a scripting language that can be used for handling complicated conditional triggering or other advanced control requirements. The user can write scripts and set them to run in response to any trigger event. From within a script you can do all the things that you can do with a trigger in the triggers screen – access passed-in variables, test conditions and perform actions - but you can also define more complicated conditional statements and perform mathematical operations.

Designer provides sample scripts which are located in Program Files at \ETC\Mosaic\Designer\resources\scripts\actions.

WARNING: Scripts are an advanced feature intended to solve problems that cannot be addressed in any other way. They are not as user-friendly as the normal triggers interface and incorrectly written scripts will not work as intended and could cause other problems with the operation of your Controller. For help with writing scripts, please see the Trigger Script Programming Guide, or please contact support to discuss requirements for a particular project.

Digital Output

The RIO 08 has eight relay outputs, the RIO 44 has four and the RIO 80 none.

Use the configuration pane to select the RIO, output and the state of the relay.

Serial Output

RS232 remains a very popular protocol for interfacing equipment and the RS232 port of a Controller or Remote Device can be configured to support most common data formats. RS485 is a more robust alternative to RS232 (better noise immunity, longer cable lengths and faster data rates) and is a widely supported protocol. A Controller or Remote Device can be configured to send RS232 full-duplex or RS485 half-duplex in the Network view, see Controller interfaces and Remote Devices. A TPC with TPC-RIO can send RS232 full-duplex.

To send serial from a Controller's serial port, use the Controller setting to specify the Controller number, leave the Device as Local and choose a port number. For the revised MSC hardware and TPC with TPC-RIO this should be set to 1. For the original MSC hardware this should be set to 1 if using the onboard RS232 port, or 2 or 3 if using an RS485 expansion module. For the MSC X this should be set to 1 or 2, depending which RS232 port is being used.

Alternatively, set the Device to a RIO and select the RIO number.

Now define the string of output characters. There are three formats in which serial strings can be entered:

Hex A series of hexadecimal characters (0-9, a-f, A-F) where pairs of values are interpreted as a byte.

Decimal A series of decimal characters (0-255) separated by "." characters.

ASCII A series of ASCII characters. The special characters '\n' for new line, '\r' for carriage return, and '\t' for tab are supported.

Ethernet Output

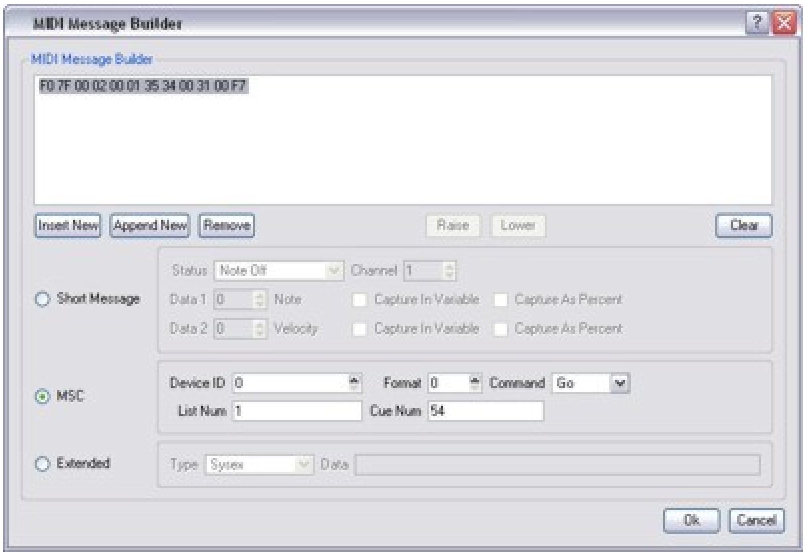
Use the Controller setting to specify which Controller should generate the Ethernet output. Define the recipient's IP address and Port, select the messaging protocol (UDP, TCP) and define the string of output characters to be transmitted. The recipients IP address and port number can be passed with variables.

MIDI Output

MIDI is another very popular protocol for interfacing equipment and the MIDI input trigger allows you to define, via a convenient MIDI Message Builder, the type (Short message, MIDI Show Control or Extended) and command string that is to be output.

Use the Controller setting to specify which MSC's MIDI port should be used as the output. To use the MIDI port on a RIO A, set the Device to RIO A and specify the RIO A number.

Press Edit to open the Message Builder:



MIDI Message Builder

MIDI Message Builder

F0 7F 00 02 00 01 35 34 00 31 00 F7

○ Short Message
Status Note Off Channel 1
Data 1 0 Note □ Capture In Variable □ Capture As Percent
Data 2 0 Velocity □ Capture In Variable □ Capture As Percent

● MSC
Device ID 0 Format 0 Command Go
List Num 1 Cue Num 54

○ Extended
Type Sysex Data

Press Insert New, select one of the three message types and then the specific command and variables.

Press Append New to add and define another command string to be output, the Raise and Lower buttons can be used to define the output order.

Press Remove to delete a command string and Clear to delete them all.

The resulting hexadecimal string will be constructed automatically and displayed in the window for reference with question marks ("??") indicating undefined characters.

Press Ok to finish.

Use Options > Preferences > Triggers to select the default Short MIDI message data format.

A comprehensive guide to MIDI is beyond the scope of this document, see the MIDI Manufacturers Association for more details, and the manual for the equipment to be interfaced will also certainly be an invaluable reference.

Set BPS Button LED

The BPS has eight buttons each with an integral white LED.

Use the configuration pane to select the BPS, button number (which can be driven by a variable) and the desired LED behaviour. Enabling "Set all other LEDs to default" will set these LEDs to their default values as specified in BPS properties.

Paradigm Action

Use this to trigger a Paradigm system. Select the type of command you wish to send then specify the parameters relevant to that action type.

Disable Audio Input

Use this to stop a RIO A or an audio expansion module from processing audio. This can aid troubleshooting as audio activity tends to fill the log. Leave Device as Local to disable a module, or select RIO A and specify the RIO A number.

Toggle eDMX Pass-Thru

Use this to enable or disable eDMX Pass-Thru on an MSC's DMX ports. Choose which port you want to enable or disable by choosing from the port selection box. See patch for more information.

Hardware Reset

Use this to force the Controller(s) to perform a hard reset which is equivalent to a power cycle. Note that unlike PC based solutions there is no particular advantage or maintenance requirement to periodically reset a Controller, this action is offered purely as a method of resetting the system to a defined, start-up state.

Triggers - TPC triggers

These triggers allow you to link the TPC's user interface with Designer's powerful show control engine and provide feedback to the TPC's controls (buttons, sliders, etc.).

TPC triggers

TPC Button

Whenever a button in a TPC user interface is touched, triggers of this type will be checked for a match. Set the Controller number to a particular TPC in order to populate the Button drop down list from the Touch Editor project (see Controller Properties).

The Button field should be set to the Control Key of the button you're interested in - this is a property of buttons that is set in Touch Editor. Either pick a control key from the list, or type it in. You can use one trigger to respond to multiple buttons by using variables - the syntax is the same as for Serial and Ethernet Input triggers, e.g. button<3d> will match a button with the control key button001 or button002, etc. and capture the number as a variable. See actions that use variables for how to make use of the captured variable.

The Event defaults to 'Click', which is a complete press and release touch action. Other options are Press, Release, Held and Repeat, like the BPS Button trigger.

TPC Slider Move

Whenever a slider in a TPC user interface is moved, triggers of this type will be checked for a match. Set the Controller number to a particular TPC in order to populate the Slider drop down list from the Touch Editor project (see Controller Properties).

The Slider field should be set to the Control Key of the slider you're interested in - this is a property of sliders that is set in Touch Editor. Either pick a control key from the list, or type it in. You can use one trigger to respond to multiple sliders by using variables - the syntax is the same as for Serial and Ethernet Input triggers, e.g. slider<3d> will match a slider with the control key slider001 or slider002, etc. and capture the number as a variable. See actions that use variables for how to make use of the captured variables.

The value of the slider will be captured as a variable. The slider value will always be the first variable - any variables captured from the control key string will be stored in subsequent variables.

TPC Colour Change

Whenever a colour picker in a TPC user interface is touched, triggers of this type will be checked for a match. Set the Controller number to a particular TPC in order to populate the Picker drop down list from the Touch Editor project (see Controller Properties).

The Picker field should be set to the Control Key of the colour picker you're interested in - this is a property of colour pickers that is set in Touch Editor. Either pick a control key from the list, or type it in. You can use one trigger to respond to multiple colour pickers by using variables - the syntax is the same as for Serial and Ethernet Input triggers, e.g. colour<3d> will match a colour picker with the control key colour001 or colour002, etc. and capture the number as a variable. See actions that use variables for how to make use of the captured variables.

The RGB colour will be captured as a variables 1, 2 and 3. Any variables captured from the control key string will be stored in subsequent variables. The default settings for the Set Fixture RGB action use variables 1, 2 and 3 to set red, green and blue, so coupling the TPC Colour Change trigger with the Set Fixture RGB action requires very little configuration.

TPC Keypad Code

When the Enter key on a keypad is pressed, triggers of this type will be checked for a match. Set the Controller number to a particular TPC in order to populate the Keypad drop down list from the Touch Editor project (see Controller Properties).

The Keypad field should be set to the Control Key of the keypad you're interested in - this is a property of keypads that is set in Touch Editor. Either pick a control key from the list, or type it in. You can use one trigger to respond to multiple keypads by using variables - the syntax is the same as for Serial and Ethernet Input triggers, e.g. keypad<3d> will match a keypad with the control key keypad001 or keypad002, etc. and capture the number as a variable. See actions that use variables for how to make use of the captured variables.

The code entered into the keypad will be captured as a variable. The keypad code will always be the first variable - any variables captured from the control key string will be stored in subsequent variables.

TPC Page Change

Whenever the current page of a TPC user interface is changed, triggers of this type will be checked for a match. Set the Controller number to a particular TPC in order to populate the Page drop down list from the Touch Editor project (see Controller Properties).

The Page field should be set to the name of the page you're interested in - this is a property of pages that is set in Touch Editor. Either pick a page name from the list, or type it in. You can also specify wether you want the trigger to fire when entering or leaving that page. You can use one trigger to respond to multiple pages by using variables - the syntax is the same as for Serial and Ethernet Input triggers. See actions that use variables for how to make use of the captured variables.

TPC Inactivity

Whenever the sleep/awake state of a TPC screen is updated, triggers of this type will be checked for a match. Set the Controller number to a particular TPC in order to trigger for the correct controller. Choose whether to trigger after a period of inactivity or when the TPC becomes active (is touched) again.

The timing for when the controller is set to inactive is in the Controller Properties area of the Network tab.

Temperature

The TPC has a temperature sensor, which can be used in triggers.

Use the Controller to specify which TPC should be considered the input source. Select the units as Celcius or Fahrenheit, then choose how to respond to changes. You can choose whether to trigger every time the temperature changes within a specified range ("Changes in range"), or to only trigger when the temperature enters a specified range ("Enters range"). "Enters range" is generally more useful when you are using temperature changes to trigger timelines, but "Changes in range" would be required if you were using the temperature reading as a variable passed to a Set Intensity action to control the intensity for a group.

TPC actions

Set TPC Control Value

Use this action to show feedback on TPC controls by changing their current value(s). Currently the Slider and Colour Picker controls support this.

Set the Controller number to a particular TPC in order to populate the Control drop down list from the Touch Editor project (see Controller Properties). Set the Control field to the target control key, or use the variable injection syntax to make this action work for several controls with similar control keys - the syntax is the same as for the Serial and Ethernet Output action.

Set the Index field to the index of the value that should be changed. For a Slider, this should always be 1, but for a Colour Picker it could be 1, 2 or 3 to set red, green or blue. The index can alternatively be set from a variable. Finally choose the value to set, or elect to set this from a variable.

Set TPC Control State

Use this action to show feedback on TPC controls by changing their appearance. The theme applied to an Touch Editor project contains various 'states' for each control type. This action lets you change the active state for a control.

Set the Controller number to a particular TPC in order to populate the Control and State drop down lists from the Touch Editor project (see Controller Properties). Set the Control field to the target control key, or use the variable injection syntax to make this action work for several controls with similar control keys - the syntax is the same as for the Serial and Ethernet Output action. It is also possible to use the wildcard character to change multiple controls at once. For example, using "button*" would set all controls with a key that begins with "button" to the specified state.

Select the state from the drop down list or choose to set the Control to its default state.

Set TPC Control Caption

Use this action to change the caption of TPC controls, including Labels.

Set the Controller number to a particular TPC in order to populate the Control drop down list from the Touch Editor project (see Controller Properties). Set the Control field to the target control key, or use the variable injection syntax to make this action work for several controls with similar control keys - the syntax is the same as for the Serial and Ethernet Output action.

Finally enter the text to set as the new caption. Variables can be used in this text, using the same syntax as for the Serial and Ethernet Output action.

Set TPC Page

Use this action to chage the current page shown on a TPC.

Set the Controller number to a particular TPC in order to populate the Page drop down list from the Touch Editor project (see Controller Properties). Set the Page field to the name of the target page, or use the variable injection syntax to make this action work for several pages with similar names - the syntax is the same as for the Serial and Ethernet Output action.

Disable TPC

The entire user interface of a TPC can be enabled or disabled. Set the Controller number to the target TPC, then choose whether to enable or disable the user interface.

Lock TPC

If TPC security has been setup in Touch Editor then this action can be used to show the lock screen on the target TPC. The user must enter the correct code on the keypad in order to unlock the TPC.

Set the Controller number to the target TPC, then choose whether to lock or unlock the user interface.

Set Screen Brightness

The brightness of the backlight of a TPC may be set using this action. Set the Controller number to the target TPC, then set the value as a percentage, or elect to set this from a variable.

NOTE: the brightness of the TPC screen can be set automatically in response to changes in ambient light - see Controller Properties.

Triggers - Conditions

Sometimes you only want a trigger to function in certain circumstances, for example during the day but not during the night. This can be achieved using the conditions, any of which can be added to a trigger and then the action will only be performed if the condition is true (or false if the "NOT" checkbox is ticked).

You may have multiple identical triggers with different conditions, allowing the same trigger to be used to do different actions - for instance start a timeline if it is not running, but stop the timeline if it is already running. You can also apply multiple conditions to a trigger and each condition must be true (or false if the "NOT" checkbox is ticked) before the action is performed.

Note that conditions do not specify a Controller - they are always tested on the same Controller as the trigger is detected. So if your trigger is a serial input on Controller 2, your digital input condition will look at the digital inputs on Controller 2. In situations where you really need to test conditions on a different Controller there are various tricks that could be used to achieve this, please contact support.

NOTE: Conditions are not tested by the simulator.

Condition types

Timeline running

Use this condition to determine if a timeline is currently running. Running is defined as being between the start and the end of the timeline - so a timeline holding at end is not running. This condition can be useful if you want to only start a timeline if it is not already running. Sometimes timelines are used as timers and this condition is used to determine if the timer has expired.

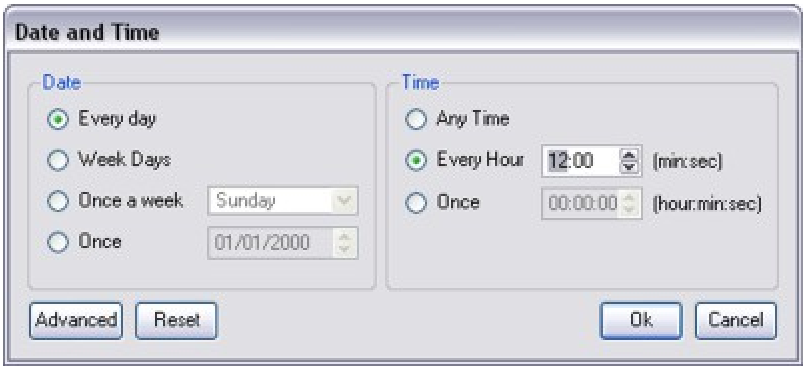
Timeline onstage

Use this condition to determine if a timeline is currently affecting the output of the Controller. It will be true as long as one fixture patched to this Controller is being controlled by the timeline. It does not matter whether the timeline is running or holding at end.

Real Time

Real time conditions can be used to limit the operation of a trigger to certain times. A single condition can be set to match if the current time is before, after or equal to the time specified. Remember that the advanced dialog can be used to set a mask - this can be particularly useful with the "equal" setting for defining ranges, for example daily opening times. Where you want to specify a very specific range of times you can use two real time conditions on the same trigger, one specifying the time it must be after and the other the time it must be before, and both must match.

The conditions work by creating a mask of times, where each value of a component (year, month, day, day of week, hour, minute or second) can either be in the mask or not. When a trigger that has this condition on it is triggered, the current time will have a single value for each component. If the operator is Equals, the mask must contain those values for the condition to be satisfied:



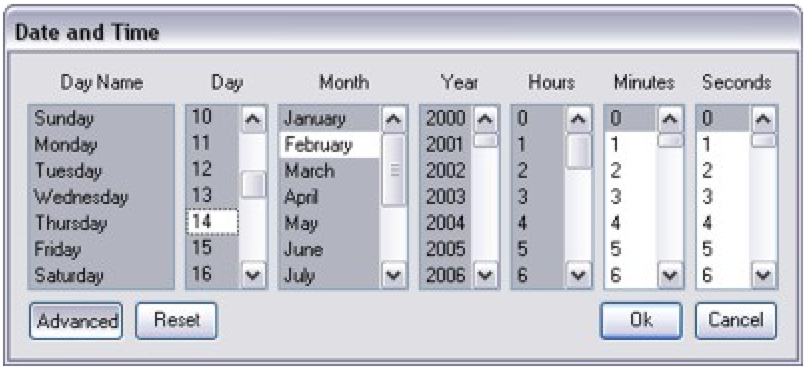
Date and Time

Date
- ● Every day
- ○ Week Days
- ○ Once a week Sunday
- ○ Once 01/01/2000

Time
- ○ Any Time
- ● Every Hour 12:00 (min:sec)
- ○ Once 00:00:00 (hour:min:sec)

Choosing "Every day" means all years, months and days are in the mask, so they will all satisfy the condition. Similarly, "Any Time" means all hours, minutes and seconds are in the mask, so any time will satisfy the condition. "Once a week" means only one day of the week is in the mask, so the condition is only satisfied when tested on that day of the week. Choosing a particular date or time means that only that date or time is set in the mask, so no other will satisfy the condition.

Using the Advanced mode, you can create more versatile masks:



Date and Time

Day Name	Day	Month	Year	Hours	Minutes	Seconds
Sunday	10	January	2000	0	0	0
Monday	11	February	2001	1	1	1
Tuesday	12	March	2002	2	2	2
Wednesday	13	April	2003	3	3	3
Thursday	14	May	2004	4	4	4
Friday	15	June	2005	5	5	5
Saturday	16	July	2006	6	6	6

For example, for the condition to be satisfied between 10pm and 4am, you would highlight all years, months, days, days of the week, minutes and seconds in the mask, but only set 22, 23, 0, 1, 2 and 3 in the hour mask. Thus, the condition will only be satisfied when the current hour is between 10pm and 4am.

Another example might be if you wanted the condition to be satisfied in every tenth minute on Sundays. Here, you would highlight all years, months, days, hours, and seconds in the mask, highlight only Sunday in the Day Name and highlight 0, 10, 20, 30, 40 and 50 in the minutes.

If the operator is set to Before (or After), the condition is satisfied if the current time is before (or after) the time set in the mask. If the mask contains a unique time (a single value for year, month, day, hour, minute and second), this should be easy to understand.

If the whole mask is set for a component of the date, that component is always satisfied as being Before (or After) the current time.

If the mask contains multiple (but not all) values in a component of the mask, only the first set value is taken. For example, if the day of the week component has Monday and Tuesday set, this is interpreted as being Before or After Monday.

When the operator is Before or After, the day of week is only considered if every value of the day component is set (so it will be satisfied on any day of the month).

NOTE: Conditions are always tested on the Controller that handles the trigger. Real time triggers are always handled on the Controller designated as Time Server. But if you use real time conditions in situations where they will be tested on Controllers that are not the Time Server then it is up to the user to make sure the time and date are set correctly on all the Controllers and not just on the Time Server although they should synchronize automatically.

Astronomical

The Controllers are also equipped with astronomical clock algorithms which automatically generate the correct sunrise, sunset and twilight times for the location of the installation (see project properties).

Astronomical conditions can be used to limit the operation of a trigger to daytime or nighttime by selecting between dawn & dusk, sunrise & sunset, sunset & sunrise, etc. You can also specify offsets, negative or positive, in minutes as required.

There is also the option to select real time instead of sunset, sunrise or twilight and enter a time to create a hybrid condition such as "Between dawn and before 01:00".

Two versions of dawn and dusk are offered, using the two definitions of twilight: *civil* and *nautical*. Please see Wikipedia for an explanation of these terms.

Note that, unlike the timeline running condition, its result may vary between different Controllers in a network system because it depends on whether fixtures in the timeline are locally patched.

Lunar

As well as astronomical triggers the Controller uses lunar clock algorithms to calculate the lunar phases based on the location of the Controller (see project properties).

Lunar conditions can be used to limit the operation of a trigger to specified lunar phases by selecting between new moon & full moon, first quarter & third quarter, etc.

Digital input

You can specify a condition based on the current state of an MSC's, TPC with TPC-RIO's or Remote Device's digital input. Leave the Device as Local to check an MSC's or TPC with TPC-RIO's input, or choose a RIO. Select the input number and whether it is active high or low (select low for contact closure).

Note that if you have more than one RIO of the same type with the same address then the condition will check against the most recent event received.

Digital inputs on Controllers or Remote Devices can also be used to detect contact closures.

Digital word

This condition allows you to test multiple of the digital inputs as a single condition. By clicking repeatedly on the numbers representing each input you can specify whether it has to be low, high or either (the default) to match.

As a side-effect the condition will also capture as a variable the state of all inputs set to match as a binary number. This can be useful if you want to pass a lot of information (such as a timeline number) using a set of digital inputs. When building the binary number low (or contact closed) is treated as a one and high (or contact open) is treated as a zero and input 1 is the least significant bit (LSB) and input 8 is the most significant bit (MSB).

Analog Input

The revised MSC hardware and TPC with TPC-RIO have 8 inputs that can be configured as digital or analog inputs in the Interfaces tab of the Network view. The RIO 80 and RIO 44 also have configurable inputs, see Remote Devices.

You can specify a condition based on the current state of an analog input. Set Device to Local to use a Controller's input or choose a RIO. Then select the input number and the percentage range of input voltage. The voltage range of an Controller's or RIO's analog input can be configured in the Network view.

DMX In

This condition will test whether the last received value for a particular DMX channel is within the specified range.

The Port setting is only necessary if using DMX-IN Expansion Modules with the original MSC hardware. In this case you can select between port 1 and 2 (in the order they are connected to the MSC).

Script

Use this to run a Lua script where the returned value determines whether the condition is true or not. Press Launch Editor to open the script editing dialog. If you can not achieve what you want with the conditions provided it is almost certain that a script can be defined to solve your problem.

The Unison Mosaic Controllers support a scripting language that can be used for handling complicated conditional triggering or other advanced control requirements. The user can write scripts and set them to run in response to any trigger event. From within a script you can do all the things that you can do with a trigger in the triggers screen – access passed-in variables, test conditions and perform actions - but you can also define more complicated conditional statements and perform mathematical operations.

Designer provides sample scripts which are located in Program Files at \ETC\Mosaic\Designer\resources\scripts\conditions.

WARNING: Scripts are an advanced feature intended to solve problems that cannot be addressed in any other way. They are not as user-friendly as the normal triggers interface and incorrectly written scripts will not work as intended and could cause other problems with the operation of your Controller. For help with writing scripts, please see the Trigger Script Programming Guide, or please contact support to discuss requirements for a particular project.

BPS Button

You can specify a condition based on the current state of a button. Select the BPS and button number and whether it is pressed (Down) or not (Up).

Remote Device Online

Use this condition to determine if a Remote Device is online (or offline with "NOT" checked), select the type (RIO or BPS) and its number.

DALI Ballast Error

Use this condition to determine if any or a single ballast(s) have reported a fixture error.

DALI Bus Power

Use this condition to determine the electrical state of a specific DALI bus.

eDMX Pass-Thru Detected

Use this to test if a valid eDMX source is detected on the specified port. See patch for more information.

Triggers - Variables

Variables are a way of collecting numbers from inputs and using them in actions. Some examples would be:

- Receiving a MIDI note on message and using the note value as a timeline number to start.
- Using a DMX input channel to master the intensity of a group of fixtures.
- Receiving a serial command on one Controller and outputting a related serial command on another.

Triggers that capture variables

▶ Timeline Started, 🟥 Timeline Ended and 🟥 Timeline Released

The Timeline Started, Timeline Ended and Timeline Released triggers capture the timeline number as variable 1 if the Timeline parameter is set to *Any*.

Digital Input

The Digital Input trigger will capture the input number if the Input parameter of the trigger is set to *Any*.

If triggering from a RIO's digital input, the trigger will capture the input number if the Input parameter of the trigger is set to *Any*. The RIO number will be captured as variable 1 and the input number as variable 2 if both these parameters are set to *Any*. If only one of these parameters is set to *Any* then the captured number will be stored as variable 1.

Analog Input

Captures the analog input as a percentage in variable 1. For example, if the input range of the Controller’s analog input is set to 0-10V and the input is 4V then variable 1 will be 40%.

If triggering from a RIO's analog input, the analog input value is captured as a percentage in variable 1, then the RIO number (if set to *Any*) and the input number (if set to *Any*) in subsequent variables. If the RIO number and the input number are set to *Any* then variable 2 will be the RIO number and variable 3 will be the input number. If the RIO number is specified then variable 2 will be the input number.

Serial and Ethernet Input

Serial and Ethernet trigger data is entered as a string of data bytes, represented in either ASCII, hex or decimal form. Any single byte or group of consecutive bytes can be matched by specifying a wildcard, and the value stored as a variable. Multiple wildcards can be used and each will store into the next available variable. There are three types of wildcards supported:

<c> or <C>

Will match any single character (or byte) and store its raw value (0-255) as the next variable. You can add a length to the wildcard to match multiple characters and treat them as a single number - so <4c> would match a 32 bit number. Maximum length = 4.

<d> or <D>

Will match a decimal character (ASCII, 0-9) and store its numeric value (0-9) as the next variable. You can add a length to the wildcard to match multiple decimal characters and treat them as a single number - so <4d> would match four decimal characters and treat them as a number from 0-9999. Maximum length = 10.

<x> or <X>	Will match a hexadecimal character (ASCII, 0-f) and store its numeric value (0-15) as the next variable. You can add a length to the wildcard to match multiple hexadecimal characters and treat them as a single number - so <2x> would match two hexadecimal characters and treat them as a number from 0-255. Maximum length = 8.
<s> or <S>	Will capture a string of arbitrary length. To determine where the string ends, you must either: • Specify a terminator yourself. For example, the trigger <s>\n would capture everything up to (but not including) the first \n character received. A terminator cannot be another variable, it must be a literal character, so <s><d> is not a valid trigger. • Send a NULL character (0x00) to the Controller to indicate the end of the string. This NULL character is assumed and is not shown in the Designer interface. You can also say that you want to capture a string with a predetermined number of characters. For example, <4s> will capture 4 bytes and store it as a string. There is no need for a terminator in this case.

Note that if the input data does not match the wildcard type then the trigger does not match. So if you have specified the wildcard <3d> and the input is ASCII "12y" then the trigger will not match because the 3 characters were not all of the required decimal type.

When using Ethernet Inputs the last two variables in the trigger will be the IP address and the source port number of the device the message was received from.

If triggering from a RIO's serial input, the RIO number will be captured as the first variable if set to *Any*.

MIDI Input

In short MIDI messages, you can capture data 1 and/or data 2 into a variable by checking the 'Capture' checkbox. If both are checked, data 1 is variable 1 and data 2 is variable 2. For some short messages, i.e. Pitch Wheel, the two data bits are treated as a single 14 bit value. To capture this 14 bit value, check 'Capture' for data 1 and check the '14 bit variable' checkbox.

In MSC messages, if the 'Cue number' and 'List number' are left blank, the received values will be captured in variables. Cue number is captured into variable 1 and list number into variable 2.

Extended messages support the same wildcard format as serial triggers. The only difference is that <2c> captures a 16-bit value in serial triggers and it captures a 14-bit value in MIDI triggers.

If triggering from a RIO A's MIDI input, the RIO A number will be captured as the first variable if set to *Any*.

DMX Input

When a DMX Input trigger matches it will implicitly store the channel value as variable 1.

Audio Input

When an Audio input trigger matches it will implicitly store the level for the band as variable 1.

If triggering from a RIO A, the RIO A number will be captured if set to *Any*.

DALI Input

If the trigger is using a Min to Max range then the matching number will be stored as variable 1.

DALI Ballast Error

If All is selected instead of a specific address then the address of the ballast reporting the error will be stored as variable 1.

BPS Button

If the button number is set to *Any*, the trigger captures the pressed button as variable 1.

Alternatively, if the BPS station number is set to *Any*, then the station number is captured as variable 1 and the button number as variable 2.

Paradigm Indicator

If any of the parameters are set to *Any* then they will be captured as variables. The variable index order will follow the order that the parameters are displayed in the trigger.

Conditions that capture variables

Digital Word

This condition will capture a variable from the inputs set to match either value. This variable will be added on the end of any variables captured by the trigger.

Conditions that use variables

DALI Ballast Error

The DALI ballast address on a specific interface can be passed in by variable. Select the relevant "Variable" option and then choose the variable index.

Actions that use variables

Captured variables can then be used in actions by specifying the variable index (corresponding to the order in which the variables were captured). If you have multiple actions associated with a trigger then each action can use the variables independently.

Start, Release, Toggle, Resume and Pause Timeline

Rather than selecting a timeline as a property of the action you can specify the timeline number in a variable. This is a very powerful feature when you want an external system to be able to call up any one of a large number of timelines because you do not need to define separate triggers for each timeline.

Set, Increase and Decrease Intensity

The intensity level or increment can be passed in by a variable. Select the "Variable" option and then choose the variable index.

Set AVC Master

The video brightness and audio volume level can be passed in by a variable. Select the "Variable" option and then choose the variable index.

Set Fixture RGB

The Red, Green and Blue values for a group of fixtures can be passed in as variables. Select the "Variable" option for the colour you want to adjust and then choose the variable index. The fade time for the action can also be passed in as a variable. Select the "Variable" option and then choose the variable index.

Set DALI Level

The DALI group or ballast number on a specific interface can be passed in by variable. Select the relevant "Variable" option and then choose the variable index.

The level of a DALI ballast, group or all DALI ballasts can be passed in by a variable. Select the "Variable" option and then choose the variable index.

Recall DALI Scene

The DALI group or ballast number on a specific interface can be passed in by variable. Select the relevant "Variable" option and then choose the variable index.

Recall a DALI scene on a specific interfaces single ballast, group or all of the ballasts. Select the "Variable" option and then choose the variable index.

DALI Command

The DALI group or ballast number on a specific interface can be passed in by variable. Select the relevant "Variable" option and then choose the variable index.

Mark DALI Ballast Fixed

The DALI ballast address on a specific interface can be passed in by variable. Select the relevant "Variable" option and then choose the variable index.

DALI Start Emergency Test

The DALI ballast address on a specific interface can be passed in by variable. Select the relevant "Variable" option and then choose the variable index.

DALI Stop Emergency Test

The DALI ballast address on a specific interface can be passed in by variable. Select the relevant "Variable" option and then choose the variable index.

Set Timeline Rate

You can select the timeline to modify with a variable (as for Start Timeline). You can also choose to pass in the rate percentage using a variable.

Serial and Ethernet Output

In the same way that you can use wildcards to match data in a serial or Ethernet trigger, you can insert the value of captured variables into your serial output messages. The same wildcard types are supported to define how to output a variable:

<c>	Will output the value of a variable as a raw byte (0-255).
<d>	Will output the value of a variable as a decimal number (ASCII, 0-9).
<x>	Will output the value of a variable as a hexadecimal number (ASCII, 0-f). Any letters will be lowercase.
<X>	Will output the value of a variable as a hexadecimal number (ASCII, 0-f). Any letters will be uppercase.
<s>	Will output a captured string. <s> will output the entire captured string. <4s> would output the first 4 characters of the captured string.

As with input you can specify a length if you want to output the variable as a longer decimal or hexadecimal number. So a variable value of 175 output with <4d> would add ASCII "0175" to the serial output. Note that it is padded with leading zeros to fill the specified length. If the value was too large to express in the specified length it would be truncated from the left, so <2d> would output the number 123 as ASCII "23".

Output strings are allowed to begin with a wildcard. By default each wildcard takes the next variable in the order they were captured. If you want to output the variables in a different order then you can add a variable index to the wildcard in the form <3,2d> where 3 is the variable index. If you specify an output wildcard where there is no corresponding capture variable then it will have value of zero and output accordingly.

MIDI Output

Short messages can output a captured value for data 1 and data 2. Pick a variable using the 'Variable Index' controls. If data 1 is outputting a captured value, you can optionally send it as a 14-bit value, with the lower 7 bits in data 1 and the upper 7 bits in data 2, by checking '14 bit variable'.

Outputting a MSC message allows you to set the cue number and/or list number by choosing a variable with the 'Variable Index' controls.

Extended messages allow you to output captured variables using the same syntax as serial actions.

{...} Run Script

Variables can be accessed from Lua scripts.

Set BPS Button LED

The button number and intensity level can be passed in by variables.

Paradigm Action

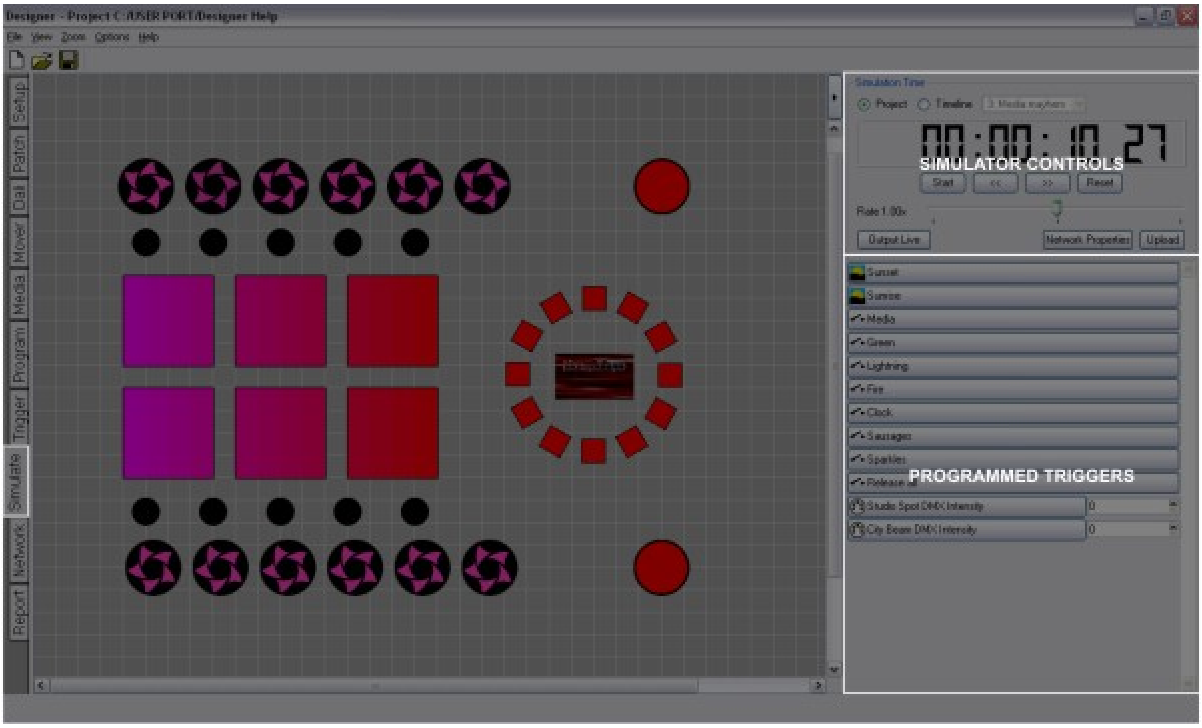
Variables can be passed into the various parameters available. Select the relevant "Variable" option and then choose the variable index.

Notes

- For hex strings, if a wildcard is inserted after an odd number of digits, the odd digit is treated as the lower 4 bits of the byte. For example, ff1<d> will be interpreted as ff01<d>.
- If you want to match a '<' character, you must precede it with a backslash. In general, a backslash followed by any character will match that character (ignoring the backslash).

Simulate

The simulator allows you to preview your programming on the plan:



SIMULATOR CONTROLS

PROGRAMMED TRIGGERS

The window comprises 3 sections: The main portion of the window is your plan. Top right are the simulator controls with time counter, and below them any programmed triggers. Note that the time counter shows simulated time not real time.

Simulator modes

Timeline mode

Use this mode to simulate a single timeline, typically the one you are editing. Select the timeline and press Start to simulate, press Reset to reset the timeline.

Project mode

Use this mode to simulate the whole presentation and verify your trigger programming and timeline interaction. Any triggers created will appear on the right hand side and can be activated by clicking on them. Activating a trigger will automatically start the simulator clock, press Reset to reset all timelines and triggers.

Simulator controls

Start/Pause

Toggles between Start and Pause accordingly, the keyboard spacebar can also be used.

Skip backwards & forwards

When the simulator is running or paused these buttons skip backwards or forwards in 10 second increments.

Reset

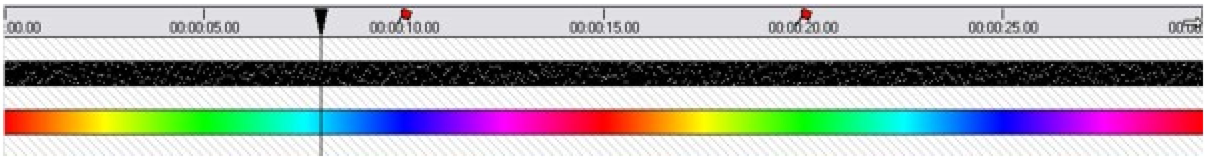
Resets the simulator, the simulated triggers (Project mode only) and releases playback so, if Output Live is enabled, the venue will go to black (fixture defaults).

Rate fader

Adjusts the simulator's playback speed (not the timeline's programming) from 0x (paused) > 1x (normal, default position) > 16x (fastest). Note that Reset does not reset this setting so remember to set it back to 1x (normal) when you've finished.

Program window play head (Timeline mode only)

In the Program window, when simulating a timeline in Timeline mode, the simulator's current playback position is marked by the play head; a black vertical line and arrow on the timeline ruler:



You can grab the play head for manual positioning by clicking on and dragging the arrow along the ruler, akin to "scratching" since you are now driving the simulator by hand at whatever speed and in whatever direction you choose, very useful for examining transitions in detail for example - easiest with the simulator paused.

You can also make the simulator jump (when running or paused) by just clicking on the ruler at the required target time, very useful for getting and keeping the simulator in an area of interest, particularly with long timelines.

Testing trigger variables

Any triggers which capture variables have a widget that allows values to be tested. These variables take the form of a comma-separated list and can either be numbers or text strings. Text strings should be bounded with double quotes.

For example, entering: 100, 50, "ABC", "100" would inject variable indices 1>4 with the values 100 (number), 50 (number), ABC (text) and 100 (text).

Testing trigger conditions

Conditions are not tested by the simulator.

Simulating timecode

By default, for any timeline that has been set to use a timecode source (see timeline properties) the simulator will emulate the timecode. If, however, a Controller is connected and receiving real timecode then the simulator can instead be made to track this real timecode by selecting Time Source from the Timecode drop-down menu.

Simulating with audio

If you need to simulate a timeline that will play in sync with an audio track in your final presentation then you can use Winamp to play the audio file and control the playback using Designer. To use this feature, you must first enable it in Preferences.

Start Winamp and cue up your audio track. In Designer's Simulator, select Winamp from the Timecode drop-down menu and start your timeline. Designer will start Winamp at the same time. Adjusting the play head in Designer will adjust the play head in Winamp automatically.

This feature is only available on Windows.

Simulating the AVC

Simulation for an AVC is shown as a thumbnail image of the main AVC playback. It will not currently show text, and all transitions are displayed as a dissolve.

Output Live

Output Live is not possible without associated and patched Controllers but this allows you to view the programming on the installation itself without having to upload repeatedly after every iteration. The Output Live button can also be found on the Mover window to facilitate the programming of the Mover Presets, particularly setting the position parameters.

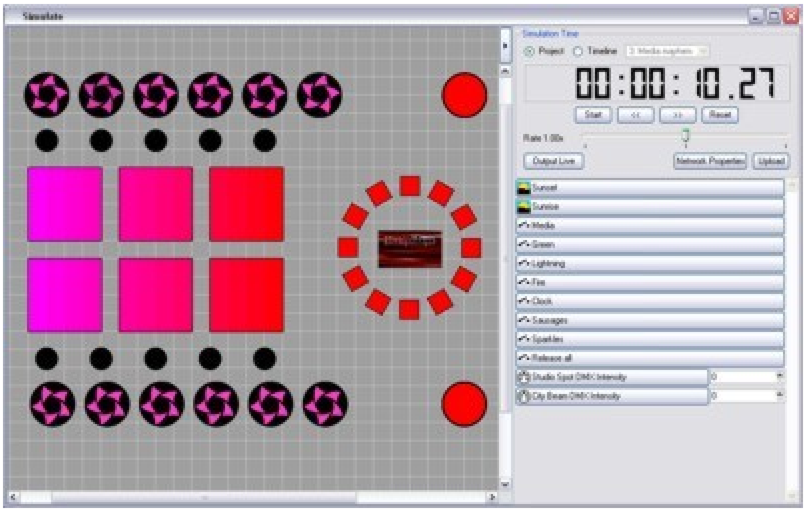
IMPORTANT: Output Live is not possible for the AVC, the project must be uploaded.

Output Live completely overrides the MSC's playback engine with all calculations instead being performed within Designer. On large installations with multiple MSCs it may not be desirable to override all of them so it is possible to exclude MSCs from participating in Output Live, see Disabling Output Live.

If you are using the tear-off simulator (see below) then Output Live will let you maintain control of the installation throughout your programming session.

Tear-off simulator

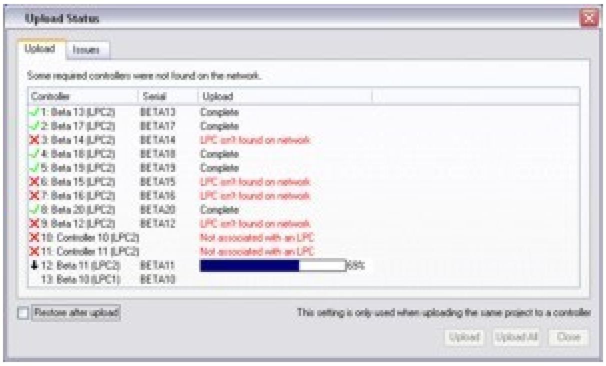
The simulator can be permanently opened for display, ideally on a second monitor:



Press Ctrl (Apple) + F8 or tear off the Simulate mode tab to open this window. Incredibly helpful in speeding up your productivity.

Upload

Once you have confirmed that your programming is as you want you can upload to the Controllers by either pressing the Upload button on the Network window or via File > Upload (Ctrl + U):



Some required controllers were not found on the network.

Controller	Serial	Upload
✓ 1: Beta 13 (LPC2)	BETA13	Complete
✓ 2: Beta 17 (LPC2)	BETA17	Complete
✗ 3: Beta 14 (LPC2)	BETA14	LPC isn't found on network
✓ 4: Beta 18 (LPC2)	BETA18	Complete
✓ 5: Beta 19 (LPC2)	BETA19	Complete
✗ 6: Beta 15 (LPC2)	BETA15	LPC isn't found on network
✗ 7: Beta 16 (LPC2)	BETA16	LPC isn't found on network
✓ 8: Beta 20 (LPC2)	BETA20	Complete
✗ 9: Beta 12 (LPC2)	BETA12	LPC isn't found on network
✗ 10: Controller 10 (LPC2)		Not associated with an LPC
✗ 11: Controller 11 (LPC2)		Not associated with an LPC
↓ 12: Beta 11 (LPC2)	BETA11	68%
13: Beta 10 (LPC1)	BETA10	

This setting is only used when uploading the same project to a controller

Direct upload (as opposed to remote, see the web interface), is not possible without connected Controllers (see the network section) but this allows you to upload your programming to one or more Controllers for stand alone operation.

You can either use the "Upload All" button to upload to all connected Controllers or select specific Controllers from list and press "Upload" to target them alone.

IMPORTANT: Changes made to a Controller's or Expansion Module's configuration will only take effect after an upload.

IMPORTANT: Changes made to the DALI configuration (including DALI groups and scenes) must be uploaded separately to the DALI ballasts, see DALI.

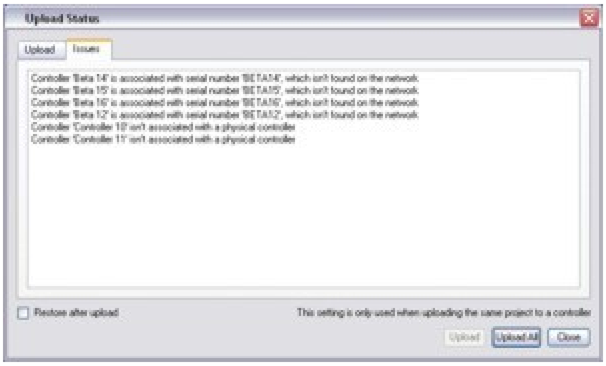
Upload to the AVC

When uploading to the AVC it may take some time to prepare media for upload and then to transfer the files over the network. Once Designer has uploaded the media files it will not need to upload them again unless you change them or upload a different project.

IMPORTANT: Currently you must connect to the AVC via Ethernet to upload content.

You may connect to the AVC via USB to modify the Controller Config (e.g. network settings) or to upload minor changes to the show.

Issues



Controller 'Beta 14' is associated with serial number 'BETA14', which isn't found on the network
Controller 'Beta 15' is associated with serial number 'BETA15', which isn't found on the network
Controller 'Beta 16' is associated with serial number 'BETA16', which isn't found on the network
Controller 'Beta 12' is associated with serial number 'BETA12', which isn't found on the network
Controller 'Controller 10' isn't associated with a physical controller
Controller 'Controller 11' isn't associated with a physical controller

□ Restore after upload

This setting is only used when uploading the same project to a controller

When you open the Upload dialog, there is a tab which lists potential errors with your project. Designer will check things like triggers and hardware configuration to make sure that there are no inconsistencies. If any issues are found, the Issues tab will be opened automatically and a description of each issue will be listed so that you can take corrective action, see Issues. You can proceed with the Upload ignoring the errors should you wish.

Restore after upload

Check the "Restore after upload" box if you want the Controllers to continue playback where they left off, useful for soft openings when programming is still being tweaked while the installation is open to the public. Note however that changes to the fixture schedule or patch will force the Controllers to reset playback and so cause a momentary black out.

What's actually uploaded?

Only the data relevant to the Controller to generate control data for the patched fixtures and utilised trigger interfaces. The uploaded data is optimised for minimal Controller memory usage and maximum playback performance.

In the case of multiple Controllers, each Controller only stores the control data required for its patched fixtures and utilised trigger interfaces and not those of the other Controllers, thus providing a scalable solution.

Can the project file be retrieved from the Controller(s)?

No, not unless you explicitly store it as an exported project (*.mde) file on the Controller's memory card. To do this you must either remove the card from the Controller and plug it into the PC or use the File Transfer feature.

Archiving a project

Export the project (File > Export or Ctrl + E) and save the *.mde file on one or more of the Controller's memory cards and keep a version for yourself safe, together with the relevant Designer installer.

IMPORTANT: If you lose the project file you will have to start from scratch.

Network - Overview

Unison Mosaic products are designed to operate on an Ethernet network for maximum scalability and the range can be split into two classes:

Controllers

The primary processing "brains" in a system, designed to operate as single, stand-alone units or co-operatively as a scalable system, automatically synchronised and managed over the network. All Unison Mosaic Controllers have an integrated web interface for remote management.

Controllers use TCP/IP for communication and, as a result, need to be correctly configured, typically via USB, see Controller connection.

Once the Controllers have been connected they are then uniquely associated with Controllers in the Designer project, see Controller association.

Tessera Panel Controller (TPC)

The TPC is an advanced lighting controller with an integrated, customisable, capacitive touch screen. It outputs up to 512 channels of eDMX and is able to output multiple protocols simultaneously. See Controller properties.

TPC with TPC-RIO

When using a TPC that is connected to a TPC-RIO the TPC functions in largely the same way. The extra output and triggering options provided by the TPC-RIO will be seamlessly applied to the TPC when the Controller Type in the Network tab reads "TPC+TPC-RIO".

Mosaic Show Controllers (MSC)

Three models are available for DMX & eDMX lighting control, the MSC 1 (512 control channels), the MSC 2 (1024 control channels) and the MSC 4 (2048 control channels). The MSC 4 can output up to 1024 channels as local DMX - the remaining channels must be output as eDMX. All 2048 channels may be output as eDMX.

The MSC X is available for eDMX and DVI control with capacities ranging from 20 universes (MSC 20; 10,240 control channels) to 200 universes (MSC 200; 102,400 control channels).

Unlike common DMX frame store devices they are extremely powerful and flexible and can be configured in numerous ways to suit the application, see Controller properties.

Audio Visual Controller (AVC)

The AVC is the solid state equivalent of two DVD players and a vision mixer, with live input, all under seamless show control. AVCs can operate stand-alone or in conjunction with one or more MSCs to provide a truly integrated lighting and audio visual control solution, see Controller properties.

Expansion Modules

The original MSC 1/2 hardware supports up to two Expansion Modules which provide additional functionality; in effect expanding the Controller's interface capabilities. Expansion Modules are thus tightly coupled to a

Controller and can not be used stand-alone nor remotely. Expansion Modules have varying configuration options, see Expansion Modules.

NOTE: The Expansion Modules are being discontinued. Their functionality is being replaced by the enhanced connectivity of the revised Mosaic Show Controller (MSC) and new Remote Devices (RIO). For further information, please contact support.

Remote Devices

The Unison Mosaic range includes various Remote Devices that augment one or more Controllers with additional remote inputs, outputs and user interfaces.

Remote Devices can only be powered by the PoE network, use multicast UDP for communication and have varying configuration options, see Remote Devices.

Network - Controller connection

Before you can configure and upload to the Controllers they must be connected to the PC running the Designer software. Depending on the Controller, this connection can generally be made in one of two ways:

USB

This is the simplest method of connecting a Controller as it avoids Ethernet TCP/IP configuration issues. However, the USB connection does not supply power to the Controller so an external 9-48V DC supply should be used and there are restrictions on USB cable length.

IMPORTANT: The TPC and MSC X can not be connected via USB.

IMPORTANT: Currently you must connect to the AVC via Ethernet in order to upload media, reload firmware or view the AVC's web interface. The AVC's USB connection may be used to configure the unit's IP settings.

Connecting the Controller via USB

Connect to your computer with a standard USB cable and, if necessary, follow the operating system's default installation procedure, though the drivers will usually be installed as part of the Designer installation.

Ethernet

This is the most flexible method of connecting one or more Controllers and is well suited to a permanent installation. TCP/IP Ethernet itself needs configuration and management, in particular the setting of IP addresses.

DHCP (default)

Unison Mosaic Controllers are factory set to receive an IP address from a DHCP server so one must be present on the network.

Link local (DHCP error)

Should the Controller fail to find a DHCP server it will assign itself a "link local" IP address which can be used with Designer to establish a temporary connection. However, a DHCP server should be found or a Static IP address set to establish a permanent connection.

Static IP (optional)

It is sometimes desirable to set a Static IP address so that the IP address of the Controller is always known (DHCP served IP addresses can change). Refer to Controller configuration.

Multicast

Unison Mosaic Controllers and Remote Devices also use a block of Multicast addresses for "discovery" and Remote Device communication so these addresses must be available: 230.0.0.0, 230.0.3.1, 230.0.3.2 & 230.0.3.3.

Default gateway

Must be consistently set to either nothing or a valid IP address.

Managed switches and firewalls

Managed Ethernet switches and your PC's Security Firewall can conspire to make life difficult - by blocking Multicast addresses for example. ETC recommends the use of Unmanaged switches and disabling your PC's Firewall if you're experiencing connection problems.

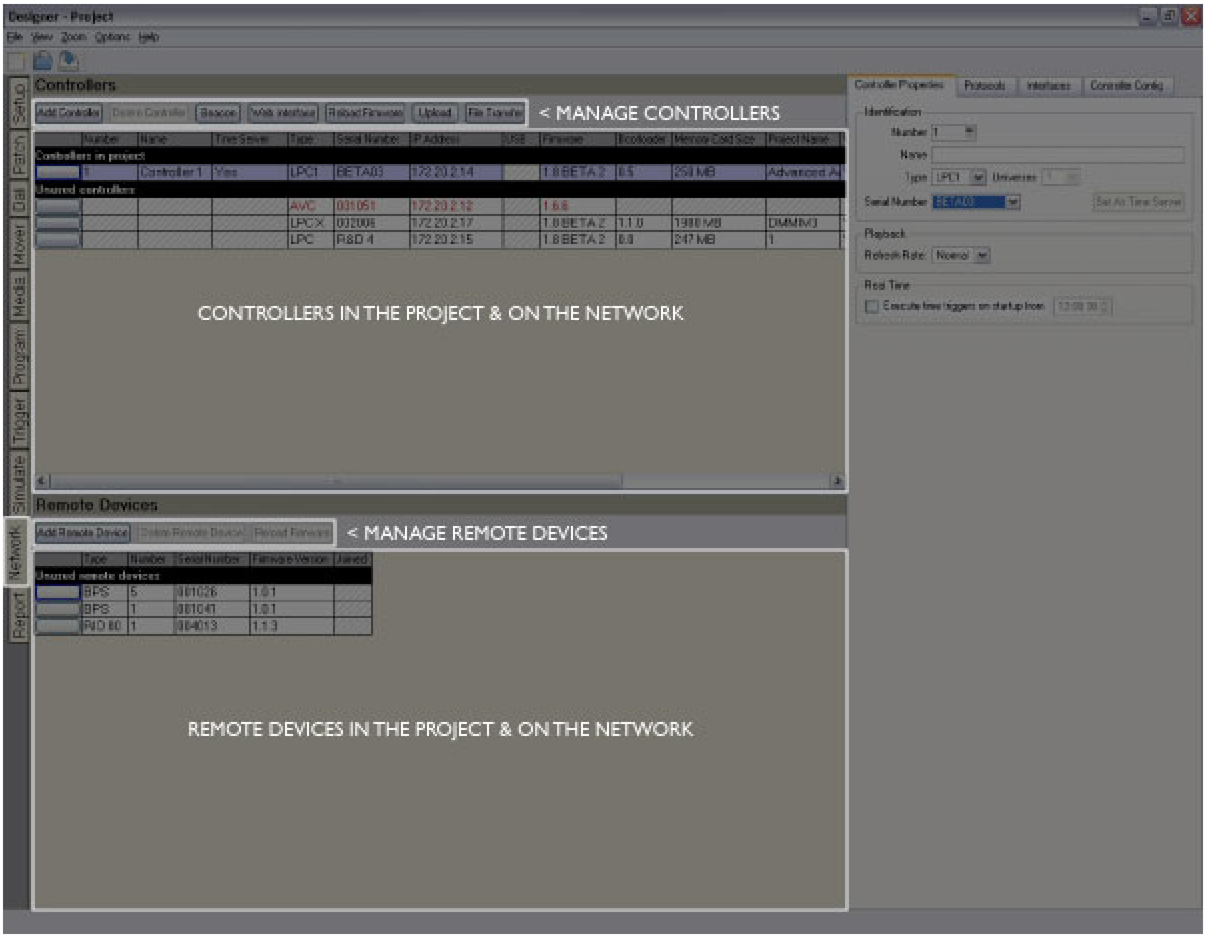
eDMX considerations

While the MSC X has a second, dedicated Ethernet Protocol port with its own IP settings, the TPC and MSC 1, 2 & 4 must share their single Ethernet port. However, this single Ethernet port can be configured with two IP addresses, making it easier to manage routing of output protocol data. See Controller Protocols.

For further information about the Controller hardware and its input/output ports please refer to the Installation Guide supplied with the unit or on the CD in PDF form.

Network window

Once you have connected the Controller, select the Network tab to view your connected Controllers & Remote Devices:



Controllers

	Number	Name	Time Server	Type	Serial Number	IP Address	USB	Firmware	Bootloader	Memory Card Size	Project Name
Controllers in project											
	1	Controller 1	Yes	LPC1	BETA03	172.20.2.14		1.8 BETA 2	0.5	250 MB	Advanced A
Unused controllers											
				AVC	031051	172.20.2.12		1.6.8			
				LPCX	032008	172.20.2.17		1.8 BETA 2	1.1.0	1988 MB	DMMM3
				LPC	R&D 4	172.20.2.15		1.8 BETA 2	0.8	247 MB	1

Identification

Number 1

Name

Type LPC1 Universes 1

Serial Number BETA03

Playback

Refresh Rate Normal

Real Time

☐ Execute time triggers on startup from 12:00:00

Remote Devices

	Type	Number	Serial Number	Firmware Version	Joined
Unused remote devices					
	BPS	5	001026	1.0.1	
	BPS	1	001041	1.0.1	
	RIO 80	1	004013	1.1.3	

If a controller is on a different network, and is not discovered on USB, it will appear as grey in the network spreadsheet. If a controller is on a different network (regardless of discovery on USB), a warning message is shown in the Controller Config tab.

Controller firmware

IMPORTANT: Controllers must be running the same version of firmware as the Designer software. Controllers with incompatible firmware will be highlighted in red.

To update a Controller's firmware:

1. Select the incompatible Controller by pressing the left hand button, the row will be highlighted
2. Press Reload Firmware on the network toolbar
3. The firmware update will proceed - you must not disturb this process

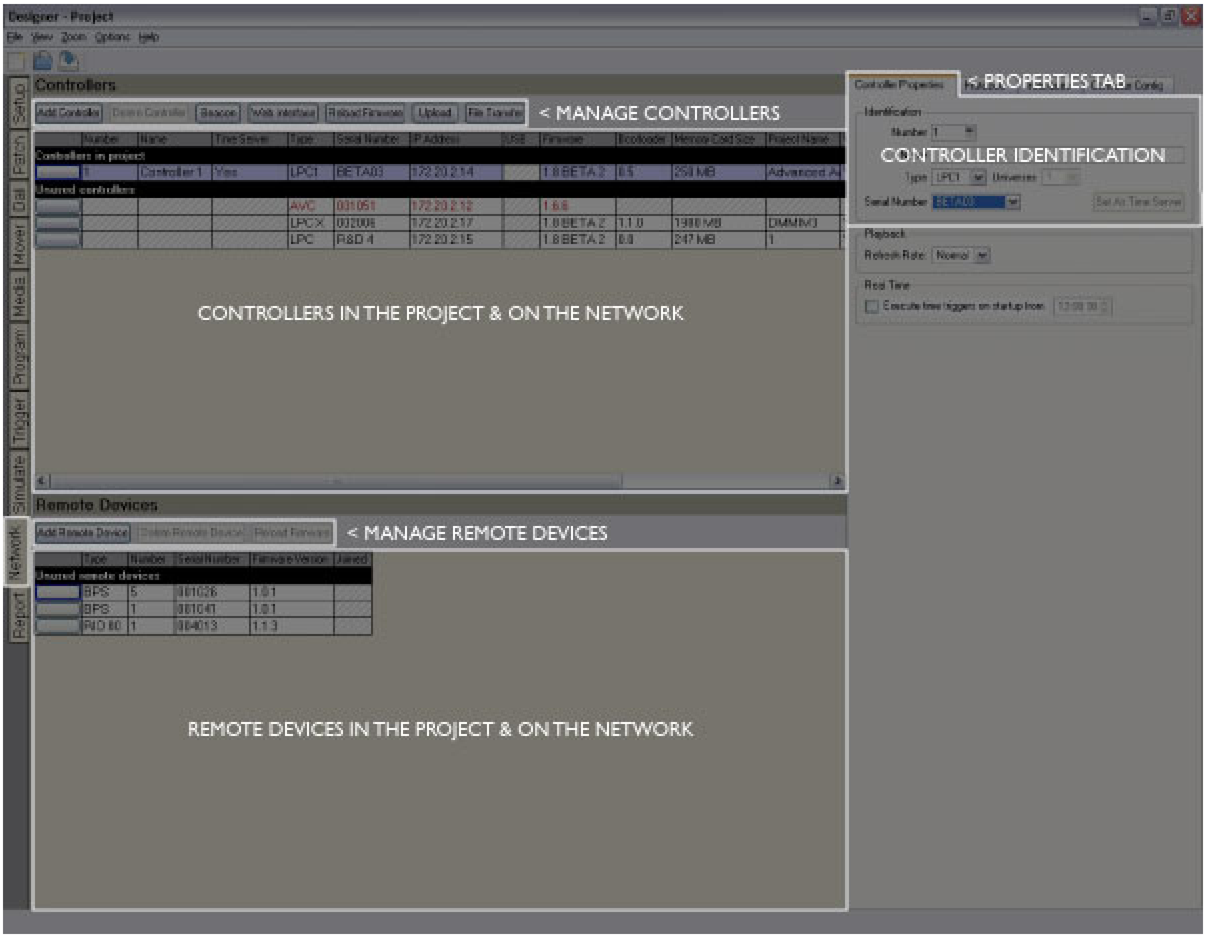
Alternatively a utility application is provided that also allows you to update the MSC X's bootloader and firmware, see MSC X Recovery Tool. There is a recovery procedure for the TPC and MSC 1/2/4.

The TPC-RIO's firmware is updated directly from the TPC. If the TPC detects that a connected TPC-RIO has the wrong firmware version then it will automatically update it - you must not disturb this process. The TPC-RIO's 'TPC Active' LED will illuminate continuously when this process has completed successfully.

Once all the connected Controllers have compatible firmware you can associate them with project Controllers and configure their hardware and any attached Expansion modules.

Network - Controller association

It is here that you connect your project programming in Designer with real, networked Unison Mosaic Controllers:



	Number	Name	Time Server	Type	Serial Number	IP Address	USB	Firmware	Bootloader	Memory Card Size	Project Name
Controllers in project											
	1	Controller 1	Yes	LPC1	BETA03	172.20.2.14		1.8 BETA 2	0.5	250 MB	Advanced A
Unused controllers											
				AVC	001051	172.20.2.12		1.6.6			
				LPC X	002008	172.20.2.17		1.8 BETA 2	1.1.0	1980 MB	DMMM3
				LPC	R&D 4	172.20.2.15		1.8 BETA 2	0.0	247 MB	1

	Type	Number	Serial Number	Firmware Version	Joined
Unused remote devices					
	BPS	5	001026	1.01	
	BPS	1	001041	1.01	
	RIO 80	1	004013	1.13	

Project vs real Controllers

The list of Controllers is split into two sections: At the top is the list of project Controllers which may or may not be associated with real Controllers. Underneath is a list of all the unused real Controllers found on the network that have not been associated with project Controllers.

When you create a new Designer project a single MSC with 2 output universes is added to get you started. These project Controllers are purely virtual and so must be associated with a real Controller on the network before you can Output Live or Upload.

IMPORTANT: You can only associate a Controller running the same firmware as Designer - Controllers running incompatible firmware will be displayed in red. To update a connected Controller's firmware see Controller firmware.

Managing project TPCs and MSCs

If you will use a different type of Controller for your project, such as a TPC for a touch screen interface or an MSC X for a high DMX channel count, or more than one Controller if the installation is large or distributed, then you will want to modify the list of project Controllers.

To add and set the type of a project TPC or MSC:

1. Press the Add Controller button on the Controller toolbar
2. Use the Controller Properties pane to give the Controller a useful name, perhaps describing where in the installation it is or what it controls
3. Use the type pull-down to select whether it is a TPC, MSC or MSC X.
4. Select the number of universes that the Controller supports (unless using the original MSC 1/2 hardware)

The Controller has now been added to the project.

If the project has a TPC with a TPC-RIO then the TPC-RIO will be configured automatically by the TPC. You can add a TPC-RIO to a TPC in project by checking the 'Configure TPC-RIO' check box in the TPC's Controller Interfaces tab.

To delete a project TPC or MSC:

1. Select the project TPC or MSC by clicking the left hand button, the row will highlight
2. Press Delete Controller on the Controller toolbar

IMPORTANT: Deleting a project TPC or MSC that has been patched will result in the loss of this patch data.

Managing project AVCs

If your project requires audio visual control, for example a plasma screen or projector, then you will want to add one or more AVCs. However, unlike MSCs, project AVCs are added or deleted only when you add or delete a generic AV fixture in Setup.

Associating project Controllers with real Controllers

Once you have added and configured your project Controllers (all that is required for programming and simulating) you must associate them with real Controllers on the network.

To associate a Controller:

1. Select the project Controller by clicking the left hand button, the row will highlight
2. In Controller Properties use the Device Serial pull-down to chose a real Controller of the same type to associate (the serial number can be found on the base of the AVC, MSC 1, 2 & 4, TPC and rear of the MSC X)
3. The real Controller will move from the Unused Controllers list and fuse with the project Controller so completing the row details

To identify a Controller (beacon):

1. Select an associated project Controller or unused Controller
2. Press the Beacon button on the Controller toolbar, all the Controller's status LEDs will flash. The screen backlight of the TPC will pulse.
3. Press Beacon again to return the Controller to normal operation

Once all your project Controllers have been associated with real Controllers you can configure them, test your programming on the installation itself and finally upload to the Controllers for stand-alone operation.

Time Server

One Controller in your project must be allocated as the Time Server, the first project Controller added is chosen by default. The Time Server is responsible for network playback synchronisation and for issuing realtime and astronomical clock triggers. To set the correct date and time set, see Controller Configuration.

To change the Time Server:

1. Select the project Controller which is to become the Time Server
2. Press the Set as Time Server button in the Controller properties pane

Web interface tools

To view a Controller's web interface:

1. Select the Controller
2. Press the Web Interface button on the Controller toolbar
3. Your PC's default browser will open the Controller's home page (or custom page if one has been created)

To create files for uploading via the web interface:

1. Use File > Save Snapshots
2. Use the file dialogue to chose a location to store the generated files
3. A file for each project Controller is created: "projectname.controllerserial.wrn" (MSCs) and "projectname.controllerserial.avc" (AVCs).

IMPORTANT: Controllers must be running the same version of firmware as the Designer software. Uploading a project file to a Controller running different firmware may result in the project failing to load and run. Check the Controller's home page to determine compatibility before attempting a remote upload.

File transfer

This feature allows you to use the Controller's memory card as a remote memory device for file storage and retrieval, typically the exported Designer project file to ensure that this remains on site with the installation for future maintenance. Note that files to be transferred in this way must adhere to the MS DOS 8.3 naming convention.

File transfer should not be confused with Upload.

To transfer files to and from a Controller's memory card:

1. Select the Controller
2. Press the File Transfer button on the Controller toolbar
3. A file browser will open to allow you to move files to and from the Controller's card, the password will be required to gain access if one has been set
4. Close the file browser to finish

Mac OS X users:

Pressing the File Transfer button in the Network pane will launch an FTP client on Mac OS X. By default, this client is Finder, and Finder doesn't allow write access to FTP servers. In order to send files to a Controller from Mac OS X, you will need to install another FTP client. We recommend Cyberduck (http://cyberduck.ch/) which is free.

When you first run Cyberduck, you should be prompted to make it the default FTP client. You should choose to do this. If you are not prompted, or Cyberduck is not used as the FTP client when attempting to perform a file transfer to a Controller, go to the Preferences in Cyberduck and under the FTP tab, there is a 'Default Protocol Handler' option. Set this to Cyberduck.

Controller status

The fields in the controller spreadsheet provide status information:

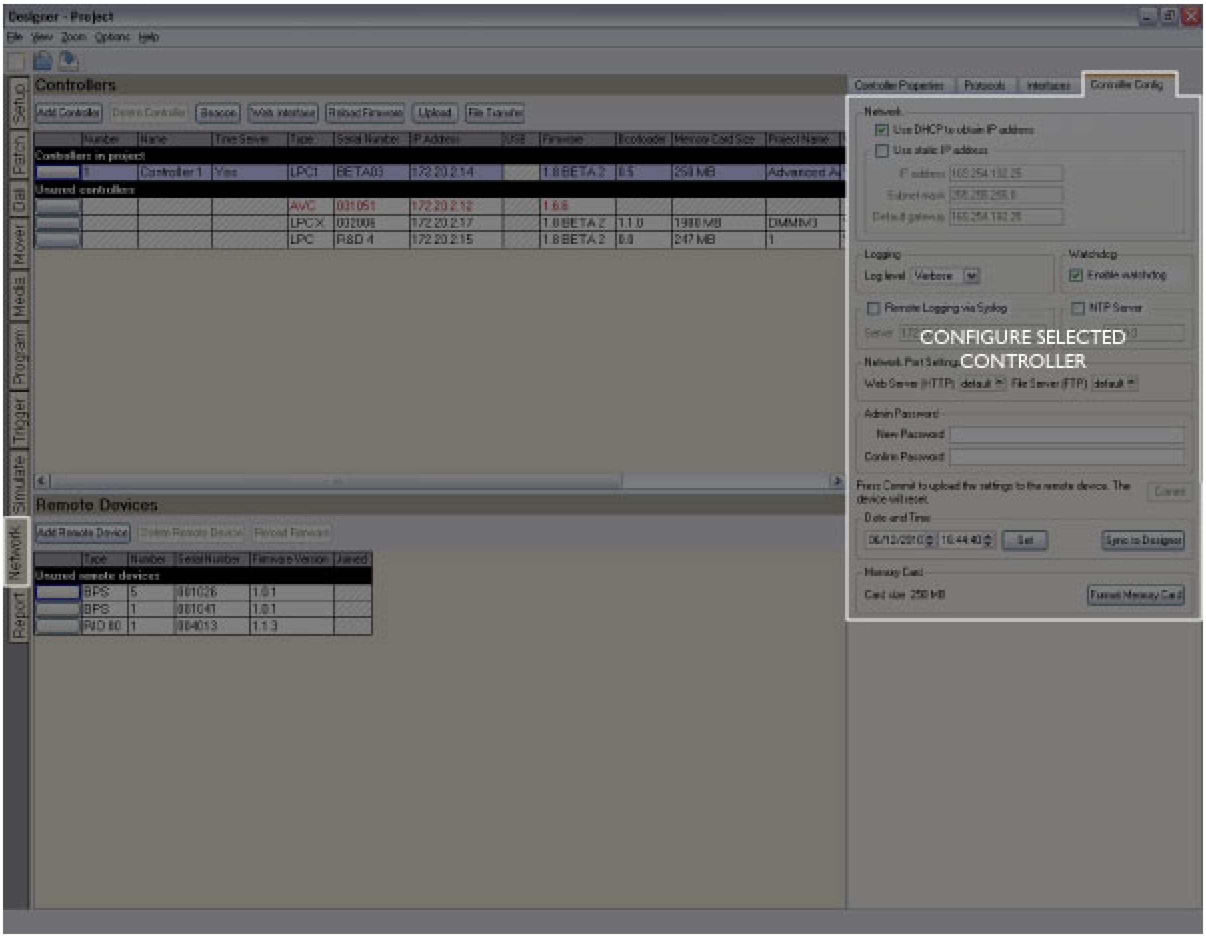
Number	The unique identifier given to each Controller in the project
Name	The user name given to each Controller in the project, typically a name that identifies the Controller's purpose or location
Time Server	"Yes" indicates that this Controller set as the Time Server
Type	The type of Controller
Serial Number	The Controller's serial number as found on the base of the Controller
IP Address	If connected via Ethernet the Controller's IP address which is either statically assigned or obtained from a DHCP server
USB	"Yes" indicates that the Controller is connected via USB
Firmware	The Controller's firmware version which must match that of Designer
Bootloader	The Controller's bootloader version
Memory Card Size	The size of the Controller's memory card, this can be user upgraded if required
Show Name	The name of the project that has been uploaded
Watchdog	"Yes" indicates that the Controller's watchdog is turned on
Used Channels	The number of used channels on the Controller
Capacity	The number of available (unused) channels on the Controller
Joined	"Yes" indicates that the Controller has joined the Designer session because it has been associated and the project files match
Output Live	Check this box (the default) if you want the Controller to participate in Output Live (see below)

Disabling Output Live

While all the other columns give status feedback, the Output Live cell is editable and allows you to stop a Controller from participating in Output Live. This is useful for soft openings where you may have multiple Controllers in various geographical zones, some of which are open to the public and so need to be protected from your programming and testing efforts in another zone. Uncheck this box to disable.

Network - Controller configuration

With a Controller selected, choose the Controller Config tab:



CONFIGURE SELECTED CONTROLLER

	Number	Name	Time Server	Type	Serial Number	IP Address	USB	Firmware	Bootloader	Memory Card Size	Project Name
Controllers in project											
	1	Controller 1	Yes	LPC1	BETA03	172.20.2.14		1.8 BETA 2	0.5	250 MB	Advanced A
Unused controllers											
				AVC	001051	172.20.2.12		1.6.6			
				LPC X	002006	172.20.2.17		1.8 BETA 2	1.1.0	1980 MB	DMMM3
				LPC	R&D 4	172.20.2.15		1.8 BETA 2	0.0	247 MB	1

	Type	Number	Serial Number	Firmware Version	Joined
Unused remote devices					
	BPS	5	001026	1.01	
	BPS	1	001041	1.01	
	RIO 80	1	004013	1.13	

Network
☑ Use DHCP to obtain IP address
☐ Use static IP address
IP address: 169.254.132.25
Subnet mask: 255.255.255.0
Default gateway: 169.254.132.25

Logging
Log level: Verbose

Watchdog
☑ Enable watchdog

☐ Remote Logging via Syslog
Server: 172

☐ NTP Server

Network Port Settings
Web Server (HTTP): default
File Server (FTP): default

Admin Password
New Password:
Confirm Password:

Press Commit to upload the settings to the remote device. The device will reset.

Date and Time
06/12/2010 16:44:40

Memory Card
Card size: 250 MB

Uniquely, these settings are stored on the Controllers themselves, not in the project or as part of the upload. They can be changed here or by using the web interface. The Controller does not have to be associated with a project Controller to do so.

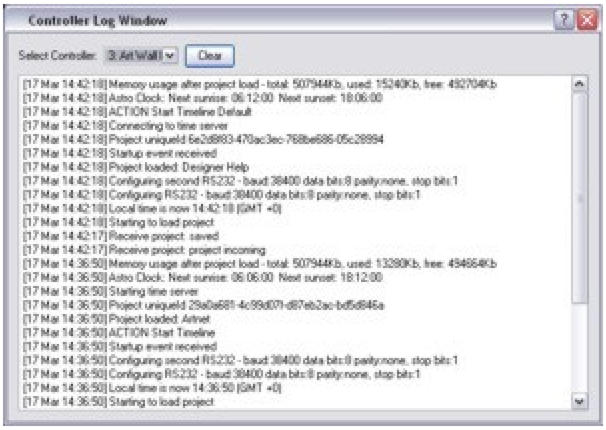
Network

Use these fields to set a static IP address for the Controller, by default the Controller is set to receive an IP address from a DHCP server.

If the IP settings have been stored on the Controller's memory card as a "TPC.cfg" or "MSC.cfg" file then these fields will be greyed out.

Logging

Select the verbosity (detail) of the log that can be viewed either via the web interface or from within Designer using View > Controller Log and selecting a Controller (which can be connected via Ethernet or USB):



```
[17 Mar 14:42:18] Memory usage after project load - total: 507944Kb, used: 15240Kb, free: 492704Kb
[17 Mar 14:42:18] Astro Clock: Next sunrise: 06:12:00  Next sunset: 18:06:00
[17 Mar 14:42:18] ACTION Start Timeline Default
[17 Mar 14:42:18] Connecting to time server
[17 Mar 14:42:18] Project uniqueId 6e2d8f83-470ac3ec-768be686-05c28994
[17 Mar 14:42:18] Startup event received
[17 Mar 14:42:18] Project loaded: Designer Help
[17 Mar 14:42:18] Configuring second RS232 - baud:38400 data bits:8 parity:none, stop bits:1
[17 Mar 14:42:18] Configuring RS232 - baud:38400 data bits:8 parity:none, stop bits:1
[17 Mar 14:42:18] Local time is now 14:42:18 (GMT +0)
[17 Mar 14:42:18] Starting to load project
[17 Mar 14:42:17] Receive project: saved
[17 Mar 14:42:17] Receive project: project incoming
[17 Mar 14:36:50] Memory usage after project load - total: 507944Kb, used: 13280Kb, free: 494664Kb
[17 Mar 14:36:50] Astro Clock: Next sunrise: 06:06:00  Next sunset: 18:12:00
[17 Mar 14:36:50] Starting time server
[17 Mar 14:36:50] Project uniqueId 29a0a681-4c99d07f-d87eb2ac-bd5d846a
[17 Mar 14:36:50] Project loaded: Artnet
[17 Mar 14:36:50] ACTION Start Timeline
[17 Mar 14:36:50] Startup event received
[17 Mar 14:36:50] Configuring second RS232 - baud:38400 data bits:8 parity:none, stop bits:1
[17 Mar 14:36:50] Configuring RS232 - baud:38400 data bits:8 parity:none, stop bits:1
[17 Mar 14:36:50] Local time is now 14:36:50 (GMT +0)
[17 Mar 14:36:50] Starting to load project
```

Watchdog

Check this to enable the internal watchdog that will reset the Controller automatically in case of a software crash as a result of either a coding error (“bug”) or a random electromagnetic event such as a power brown-out or spike, nearby lightning strike or static discharge. A startup trigger will be required to determine what the Controller should do after such a reset, see triggers.

Remote Logging via Syslog

Check this to enable logging to the specified IP address. Note that there is a performance penalty to pay for using Syslog so this should only be enabled for debugging.

NTP Server

Check this and enter the IP address of the appropriate Network Time Protocol (NTP) server. Note that Controllers with DHCP enabled will also synchronise with a suitably configured DHCP server. The AVC does not currently support NTP.

Network Ports

The ports opened by the Controller for access to the web server using HTTP and for access to the FTP server for file transfer can be manually configured. This can be useful if there are several Controllers in an installation and remote access is required via a router setup for port forwarding to each Controller.

By default the Controller uses port 80 for the web server and port 21 for the FTP server.

Admin Password

Enter a password to protect the Controller from unauthorised access. Once a password has been set it will be required to Upload, Reload Firmware, change the Controller Configuration (these settings) or open the Control and Configuration sections of the web interface.

To change the Controller configuration settings:

1. Make the required changes to any of the fields as described above
2. Press Commit, the settings are then stored on the Controller itself (they are not stored in the project)

3. The Controller will reset

Date and Time

All Controllers have an internal realtime clock which is battery backed and so will operate even when the Controller is not powered. Whilst the internal realtime clock is accurate, the use of a Network Time server of some sort (NTP, DHCP) is recommended where possible (see above).

The Date and Time fields display the current settings of the selected Controller's realtime clock. Only the designated Time Server need be accurate as any other networked controllers will automatically synchronize their realtime clocks to the Time Server.

Note that the project location settings include the correct GMT offset so, if using these location settings, you should set the time here to GMT not local time or the offset will be doubled.

To change manually the Time Server's date and time:

1. Select the designated Time Server
2. Enter the required settings into the Date and Time fields
3. Press Set

To synchronize the Time Server's date and time to Designer:

1. Select the designated Time Server
2. Ensure that the PC running Designer is set to the correct date and time
3. Press Sync to Designer

Memory Card

The capacity of the Controller's memory card is displayed here with the option to format the card if required, press "Format Memory Card". Formatting the card will erase all project data and so an upload will be required to restore normal operation.

Since the AVC uses the card aggressively, best performance requires the memory card be formatted in a particular way. It is therefore recommended that you format the memory card when starting a fresh AVC project or when inserting a new card.

Storing configuration settings on the memory card (optional)

After a reset, MSCs look for a file called "MSC.cfg" on the memory card before using its current settings. TPCs look for a file called "TPC.cfg". You can use a text editor (e.g. Notepad) to create this file and copy it to the memory card to force the issue, useful for transferring the IP settings on the card with the project data. The format of the file needs to be:

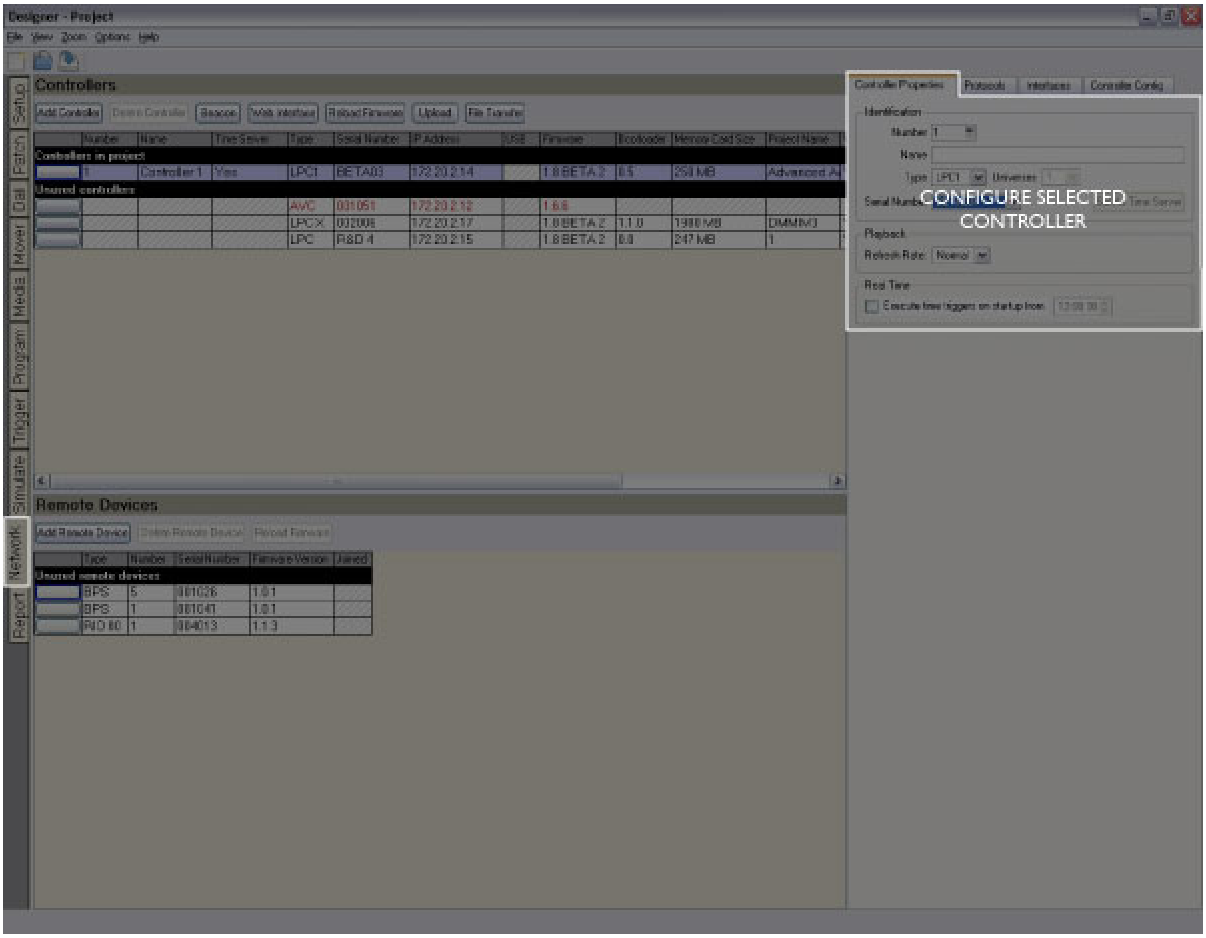
```
ip 192.168.42.56 255.255.255.0 192.168.42.250
ntp 192.168.42.1
syslog 192.168.42.8
loglevel 3
```

The first line is the IP Address, Subnet Mask and Default Gateway. The other lines are optional and define the NTP and SYSLOG IP addresses and the LOGLEVEL setting.

Using the TPC.cfg or MSC.cfg file to store the Controller's configuration on the memory card allows a Controller to be swapped, in case of failure for example, by just moving the memory card into another Controller.

Network - Controller properties

With a Controller selected, choose the Controller Properties tab:



	Number	Name	Time Server	Type	Serial Number	IP Address	USB	Firmware	Bootloader	Memory Card Size	Project Name
Controllers in project											
	1	Controller 1	Yes	LPC1	BETA03	172.20.2.14		1.8 BETA 2	0.5	250 MB	Advanced A
Unused controllers											
				AVC	001051	172.20.2.12		1.6.6			
				LPC X	002006	172.20.2.17		1.8 BETA 2	1.1.0	1980 MB	DMMM3
				LPC	R&D 4	172.20.2.15		1.8 BETA 2	0.0	247 MB	1

	Type	Number	Serial Number	Firmware Version	Joined
Unused remote devices					
	BPS	5	001026	1.01	
	BPS	1	001041	1.01	
	RIO 80	1	004013	1.13	

Identification

Number 1

Name

Type LPC1 Universes 1

Serial Number

Playback

Refresh Rate: Normal

Real Time

□ Execute time triggers on startup from 12:00:00

Identification

Use these fields to identify a project Controller with a name and type, then associate it with a real Controller and set the Time Server, see Controller Association.

Screen

TPC

User interfaces for the TPC are created using the companion Touch Editor application, which is available to download from our website. Browse to the Touch Editor project file to associate it with the selected TPC. Changes to the Touch Editor project file will automatically be detected. To remove the current Touch Editor project file, click the clear button.

Set the backlight brightness for normal operation and for when the TPC has been inactive for a period of time. The inactivity time is also set here, along with the time before the screen turns off completely.

Set whether the backlight brightness should automatically adjust for changes in the ambient light level, and whether the screen should turn on if the proximity sensor detects someone walking up to it.

AVC

AV devices can be configured to set the output format for your connection to the device ("Composite/S-video" or "Component") as well as the input format. Choose a background colour for the video screen (default is black).

In situations where it is necessary to lock the refresh timing of the video screen to be the same as another video signal then you should connect the other video signal to the appropriate AVC's video input and select the "Genlock enabled" option.

IMPORTANT: Genlock is only required in very specific circumstances - such as when the screen is being viewed by cameras - and it should be left unselected in all other situations.

The AVC's default audio level can be set here, the default is main row audio at full (255).

Playback Refresh Rate (TPC and MSC only)

Select between *Normal* (33Hz: default, recommended), *High* (44Hz) or *Low* (20Hz: useful for older fixtures with DMX compatibility issues).

Real Time

If your project uses realtime or astronomical triggers then, when the Controller starts up, you may well want to assert the correct show state for the current time. You can set each Controller to do this automatically by checking the “Execute realtime triggers on startup” option.

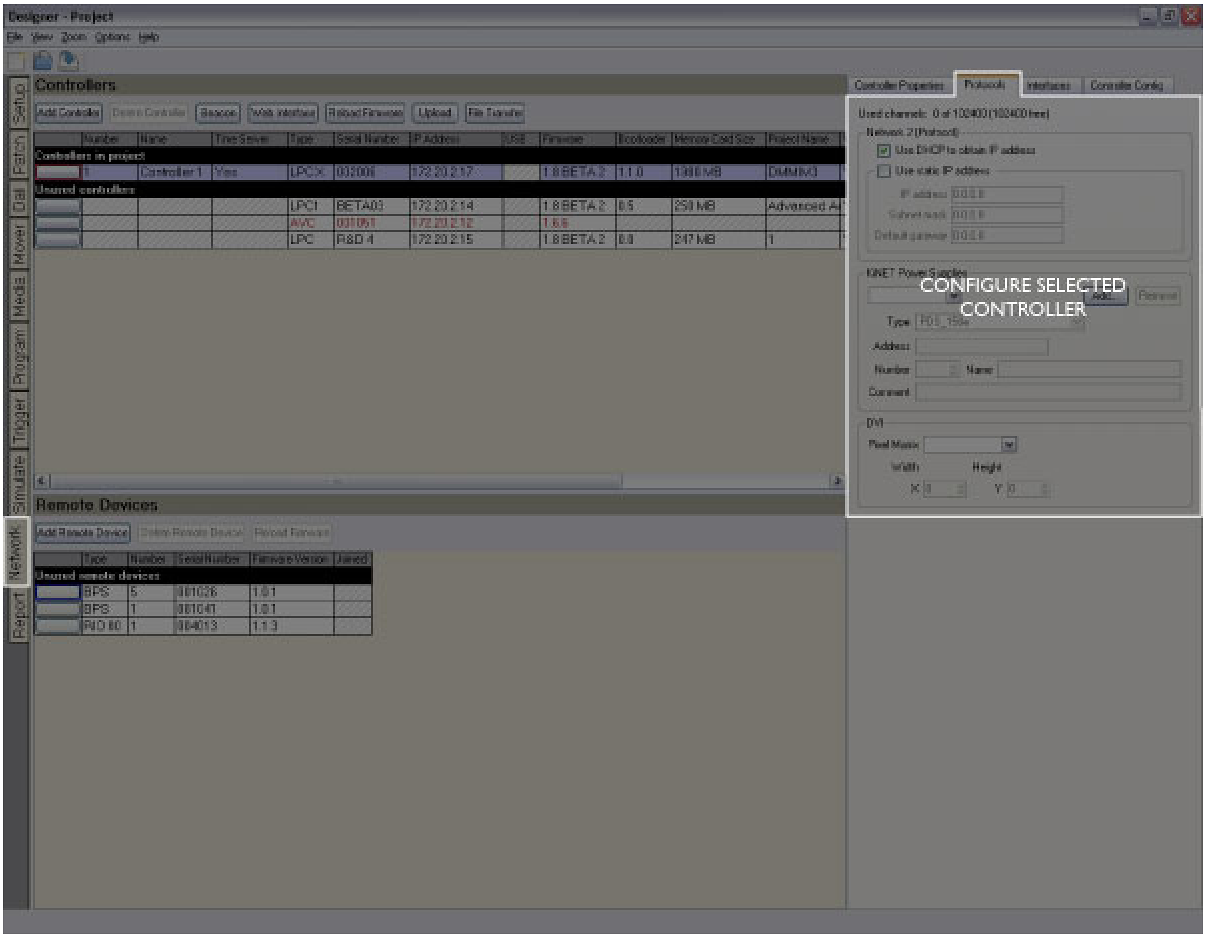
When this option is selected the Controller will execute all the realtime and astronomical triggers that would have fired between the user-specified time and the current time. You should select a time of day when your project is inactive or at its least active. Running all triggers from that point ensures that your project is in the correct state.

Any startup trigger in your project will run first and then any pending realtime or astronomical triggers. The Controller will attempt to preserve the timing so that timelines will be in the correct place as if it had been running normally.

IMPORTANT: Changes made to a Controller's properties will only take effect after an upload.

Network - Controller protocols

With a Controller selected, choose the Protocols tab which is also available from the Patch window:



	Number	Name	Time Server	Type	Serial Number	IP Address	USB	Firmware	Bootloader	Memory Card Size	Project Name
Controllers in project											
	1	Controller 1	Yes	LPC X	003006	172.20.2.17		1.8 BETA 2	1.1.0	1886 MB	DMMIX3
Unused controllers											
				LPC1	BETA03	172.20.2.14		1.8 BETA 2	0.5	250 MB	Advanced A
				AVC	001051	172.20.2.12		1.6.6			
				LPC	R&D 4	172.20.2.15		1.8 BETA 2	0.8	247 MB	1

Used channels: 0 of 102400 (102400 free)

Network 2 (Protocol)

☑ Use DHCP to obtain IP address

☐ Use static IP address

IP address: 0.0.0.0

Subnet mask: 0.0.0.0

Default gateway: 0.0.0.0

KiNET Power Supplies

Type: PDS_150e

Address:

Number: Name:

Comment:

DVI

Pixel Matrix:

Width Height

X: 0 Y: 0

	Type	Number	Serial Number	Firmware Version	Joined
Unused remote devices					
	BPS	5	001026	1.01	
	BPS	1	001047	1.01	
	RIO 80	1	004013	1.13	

Depending on the Controller type selected, the Protocols tab allows you to configure the available eDMX and other output protocols:

Network 2 (Protocol)

TPC, revised MSC 1, 2 & 4

Though the TPC and MSC 1, 2 & 4 (revised hardware) only have a single Ethernet port, this port may be configured with two IP addresses - one for management data and the other for output protocol data. The default setting is for this dual IP mode to be disabled, so that the same IP settings are used for management and output protocol data. Dual IP mode is particularly useful for working with KiNet power supplies, which must use 10.xx.xx.xx addresses.

MSC X

The MSC X has a dedicated Ethernet Protocol port. The default setting is to obtain IP settings via DHCP but static IP settings can alternatively be set as required.

DMX Proxy (MSC 1 only)

If a Designer project has a TPC and an MSC 1, the TPC can output local DMX via the MSC's second DMX port. With the MSC 1 selected, choose the TPC from the drop down list.

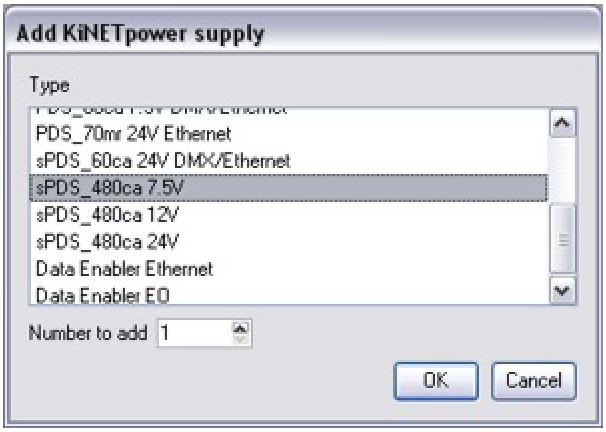
ETCNet2

Configure as required:

- Source number - ETCNet2 multicast address references, from 1 to 26, default is 1
- Source priority - ETCNet2 output arbitration references, from 1 to 20, typically 10

KiNet power supplies

These configuration settings are used to manage the KiNet power supplies if required. Press the Add button to add a KiNet power supply using the Select a power supply dialog:



Add KiNETpower supply

Type: sPDS_480ca 7.5V

Number to add: 1

Select the appropriate power supply from the list and press Ok (or Cancel to abort). Add more power supplies if required.

Once you have added your KiNet power supplies use the power supply pull-down and the fields below to configure each in turn:

- Address - The power supply's IP address
- Number - A unique user number for the power supply
- Name - A name for the power supply
- Comment - An optional field for additional information (location, purpose etc)

Use the Remove button to delete the selected power supply, all patch data will be lost.

DVI (MSC X only)

To output data using the DVI port you must first create a pixel matrix that matches the LED controller's pixel map. Once this has been done, use the Pixel Matrix pull-down on the protocol tab to select which matrix will be output via the DVI port. Any programming for the fixtures in the pixel matrix will now output on the DVI port, not just programming applied directly to the pixel matrix.

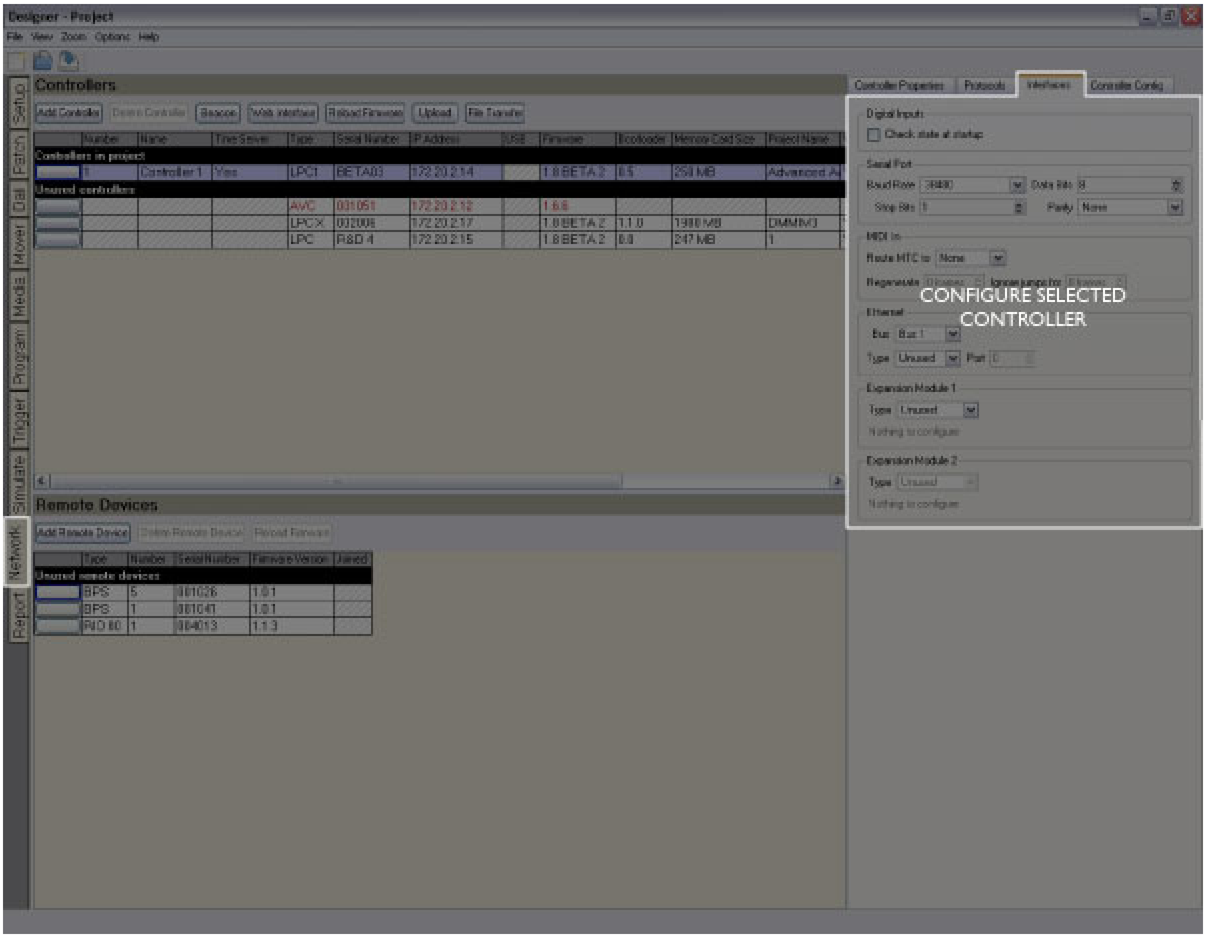
The MSC X's DVI port is set to a fixed 1024x768@60Hz resolution which is compatible with most LED controllers. The LED controller (or monitor) MUST be connected when the MSC X boots or resets for the port to become active.

The X and Y offset of the pixel matrix within the 1024x768 DVI output can be set as required. The size of the pixel matrix can be scaled up using the multiplier setting - each pixel in the matrix will occupy an area equal to the square of the multiplier on the DVI output.

IMPORTANT: Changes made to a Controller's protocols will only take effect after an upload.

Network - Controller Interfaces

Choose the Interfaces tab to configure the input/output interfaces for a Controller, including any Expansion Modules:



	Number	Name	Time Server	Type	Serial Number	IP Address	USB	Firmware	Bootloader	Memory Card Size	Project Name
Controllers in project											
	1	Controller 1	Yes	LPC1	BETA03	172.20.2.14		1.8 BETA 2	0.5	250 MB	Advanced A
Unused controllers											
				AVC	001051	172.20.2.12		1.6.6			
				LPC X	002006	172.20.2.17		1.8 BETA 2	1.1.0	1980 MB	DMMM3
				LPC	R&D 4	172.20.2.15		1.8 BETA 2	0.0	247 MB	1

	Type	Number	Serial Number	Firmware Version	Joined
Unused remote devices					
	BPS	5	001026	1.0.1	
	BPS	1	001041	1.0.1	
	RIO 80	1	004013	1.1.3	

Digital Inputs
□ Check state at startup

Serial Port
Baud Rate: 38400 Data Bits: 8
Stop Bits: 1 Parity: None

MIDI In
Route MTC to: None
Regenerate: Disabled Ignore jumps for: Disabled

Ethernet
Bus: Bus 1
Type: Unused Port: 0

Expansion Module 1
Type: Unused
Nothing to configure

Expansion Module 2
Type: Unused
Nothing to configure

IMPORTANT: Changes made to a Controller's interfaces will only take effect after an upload.

Configure TPC-RIO (TPC only)

Check the "Configure TPC-RIO" box if you want to see TPC-RIO interfacing options for a TPC.

Inputs (MSC 1, 2 & 4 and TPC with TPC-RIO only)

Check the "Check State At Startup" box if you want the inputs to be read and acted upon in triggers at startup.

The inputs of the revised MSC and TPC-RIO can be individually configured as either Contact Closure, Digital or Analog with the latter two modes allowing for the threshold or range to be selected. The maximum voltage range is 0-24V and the smallest measurable change is 0.25V.

Serial Port(s) (not standalone TPC)

Use these fields to configure the Controller's integrated RS232 or RS232/485 serial port(s) specifying the baud rate, the number of data and stop bits as well as any parity bits used to match the settings of the connected device.

Note that the Controllers do not use a specific serial protocol but instead can generate or match any serial string by setting up the appropriate triggers. That being said, the serial port on the revised MSC 1/2/4 hardware and the AVC can be configured here to receive the DMX protocol directly.

MIDI In (MSC 1, 2 & 4 only)

The MSC's MIDI Input can read MIDI Timecode (MTC) allowing a presentation to be synchronised with audio-visual or show control equipment. The configuration options are:

- Route To - select one of the six Timecode Buses to which the MIDI timecode (MTC) should be routed.
- Regenerate for - select the number of frames that should automatically be generated in the case of loss of a valid signal.
- Ignore jumps for - select the number of frames that should be considered a valid jump in the timecode value.

MIDI messages other then MTC, for example MIDI Notes or MIDI Show Control (MSC), require no configuration and these protocols can be used simply by setting up the appropriate triggers.

Ethernet

The Controller's Ethernet port can send and receive Ethernet messages allowing a presentation to be synchronised with Building Management Systems (BMS) or show control equipment. The configuration options are:

- Route To - select one of the five Ethernet Buses to which the incoming Ethernet messages should be routed.
- Type - select the messaging protocol (UDP, TCP or Multicast - the latter will require an IP address).
- Port - enter the appropriate port.

Note that the Controllers do not use a specific Ethernet protocol but instead can generate or match any Ethernet string by setting up the appropriate triggers.

DMX-In (MSC X and TPC only)

The MSC X has an integrated DMX Input port but it also supports DMX Input via Art-Net II or sACN. Select either DMX for the integrated port or Art-Net II or sACN and a Universe number for eDMX reception. Note that the Controller only listens for Art-Net on its Protocol port.

The TPC also supports DMX Input via Art-Net II and sACN. Select either Art-Net II or sACN and a Universe number for eDMX reception.

DALI

The TPC-RIO has a DALI bus interface. This can be used to control DALI ballasts via timeline programming and to receive DALI commands for use in DALI Input triggers.

MSC 1 & 2: Expansion Modules

A variety of Expansion Modules are available to augment the original MSC 1 & 2 hardware, though these are being discontinued. Their functionality is being replaced by the enhanced connectivity of the revised MSC and new Remote Devices. For further information, please contact support.

Up to two Expansion Modules may be connected to each MSC in a project. Please refer to the documentation supplied with the modules for hardware details and installation instructions.

IMPORTANT: Changes made to an Expansion Module's configuration will only take effect after an upload.

Options vary according to the module type, Designer v1.11 supports the following:

Paradigm

To enable a controller to talk to a Unison Paradigm Processor, specify the IP address here. See advanced triggers for further information on Paradigm integration.

DMX-IN

The TPC and MSC X support Art-Net and sACN input. Select the Art-Net or sACN universe to listen to.

For the original MSC 1 & 2 hardware, the DMX Input Expansion Module allows one or more Controllers to be controlled by any USITT DMX512 compatible system. There are no configuration options.

In both cases, see advanced triggers and conditions for usage.

RS485

The RS485 Expansion Module allows one or more Controllers to receive or transmit RS485 serial command strings to facilitate interfacing to show control and user interface devices. The configuration options are:

- Receive/Transmit - check the appropriate box as required.
- Baud rate - select the baud rate.
- Data bits - select the number of data bits (typically 8).
- Stop bits - select the number of stop bits.
- Parity - select the parity type.

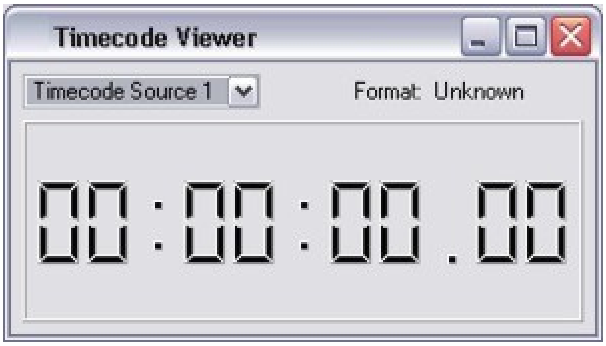
See advanced triggers and conditions for usage.

LTC

The LTC Expansion Module reads all Linear Timecode (LTC) formats allowing a presentation to be synchronised with audio-visual or show control equipment. The configuration options are:

- Route To - select one of the six Timecode Buses to which the timecode should be routed.
- Regenerate for - select the number of frames that should automatically be generated in the case of loss of a valid signal.
- Ignore jumps for - select the number of frames that should be considered a valid jump in the timecode value.

The Timecode Viewer is available from the View menu to monitor each Timecode Bus:



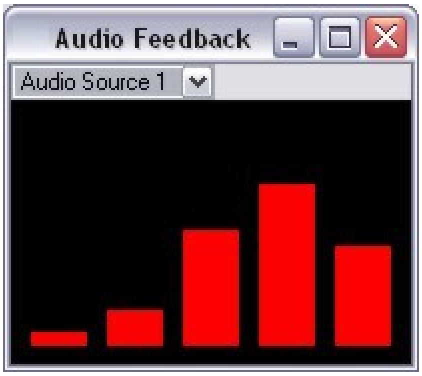
See timeline properties for usage.

AUDIO

The Audio Expansion Module allows realtime manipulation of intensity and timeline playback to create "sound-to-light" or "light organ" effects. The configuration options are:

- Route To - select one of the four Audio Buses to which the audio should be routed.
- Gain - adjust the module's input gain to suit the audio signal level.
- Initially Enabled - uncheck to disable this port.

The Audio Feedback window is available from the View menu to monitor each Audio Bus:



The Audio Expansion Module provides five frequency bands at 63Hz, 250Hz, 1KHz, 3.5KHz and 12KHz.

The RIO A has superseded the Audio Expansion Module and offers vastly superior functionality. See remote devices for more details.

See timeline properties and advanced triggers for usage and please contact support for tips and tricks as a surprising variety of effects are available.

DALI-M

The DALI Master Expansion Module allows control of up to 64 DALI ballasts, with each module being mapped to a DALI Interface, see DALI. Since patch, ballast configuration, group and scene data is stored in the DALI ballasts themselves it is beneficial to be connected to the ballasts when programming.

Use the Interface pull-down to select which DALI interface the Module is assigned to. Only one Module can be assigned to each interface of which there can be up to 4, see DALI for adding and removing DALI interfaces.

DALI-S

The DALI Slave Expansion Module allows one or more Controllers to be controlled by any DALI compatible system, see advanced triggers for usage. The Module snoops the DALI bus and does not, typically, require a ballast address. However, some controllers may not function unless ballasts are detected in which case use the "Number Of Ballasts" setting to have it emulate one or more ballasts.

Network - Remote Devices

Please refer to the documentation supplied with the units for hardware details and installation instructions.

Connection

Remote Devices can only be connected using a Power over Ethernet (PoE) connection so a suitable PoE repeater or switch must be provided.

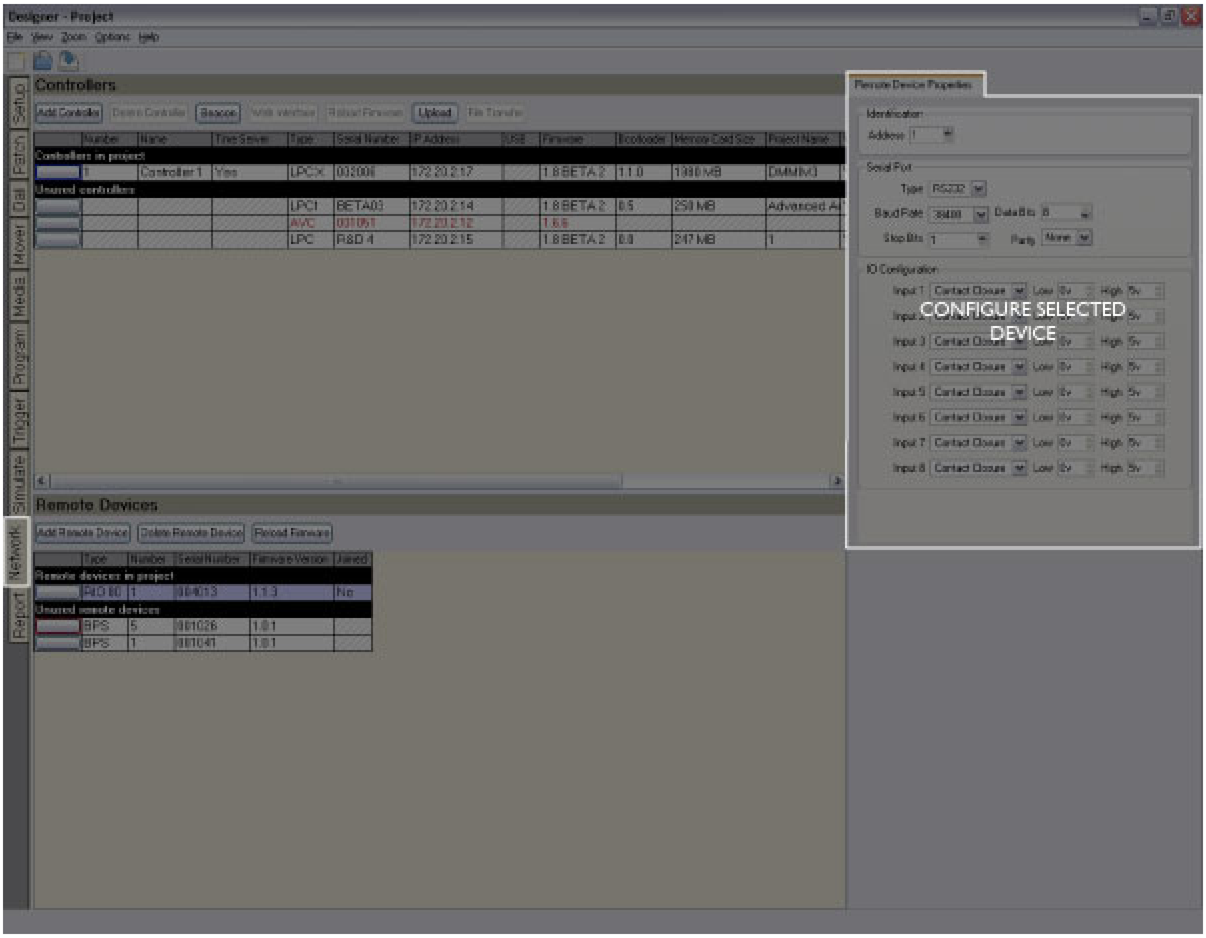
TCP/IP

Since the Remote Devices do not use TCP/IP, connection is more straightforward than with the Controllers, simply connect them to the network and use the thumb wheel on the device to set an Address, see Associating Remote Devices below.

Multicast

Remote Devices do however use a block of Multicast addresses for communication so these addresses must be available: 230.0.0.0, 230.0.3.1, 230.0.3.2 & 230.0.3.3. Please be aware that managed Ethernet switches which may block these addresses by default.

The bottom half of the Network window allows you to manage and configure any Remote Devices in the project and found on the network:



	Number	Name	Time Server	Type	Serial Number	IP Address	USB	Firmware	Bootloader	Memory Card Size	Project Name
Controllers in project											
	1	Controller 1	Yes	LPCX	003006	172.20.2.17		1.8 BETA 2	1.1.0	1980 MB	DMMM3
Unused controllers											
				LPC1	BETA03	172.20.2.14		1.8 BETA 2	0.5	250 MB	Advanced AI
				AVC	001051	172.20.2.12		1.6.6			
				LPC	R&D 4	172.20.2.15		1.8 BETA 2	0.8	247 MB	1

	Type	Number	Serial Number	Firmware Version	Joined
Remote devices in project					
	RIO 80	1	004013	1.1.3	No
Unused remote devices					
	BPS	5	001026	1.01	
	BPS	1	001041	1.01	

Identification
Address: 1

Serial Port
Type: RS232
Baud Rate: 38400 Data Bits: 8
Stop Bits: 1 Parity: None

IO Configuration

Input 1	Contact Closure	Low	0v	High	5v
Input 2	Contact Closure	Low	0v	High	5v
Input 3	Contact Closure	Low	0v	High	5v
Input 4	Contact Closure	Low	0v	High	5v
Input 5	Contact Closure	Low	0v	High	5v
Input 6	Contact Closure	Low	0v	High	5v
Input 7	Contact Closure	Low	0v	High	5v
Input 8	Contact Closure	Low	0v	High	5v

Project vs real Remote Devices

The list of Remote Devices is split into two sections: At the top is the list of project devices which may or may not be associated with real devices. Underneath is a list of all the unused real devices found on the network that have not been associated with project devices.

Managing project Remote Devices

To add and set the type of a project Remote Device:

1. Press the Add Device button on the Remote Device toolbar
2. In the Add Remote Device dialog, select the device type (RIO 80, RIO 44, RIO 08, RIO A, RIO D, BPS)
3. Choose the device's number (the address selected on the device itself, see Associating Remote Devices below)
4. Choose the device's controller.
5. Press OK, the Remote Device will be added to the project (and associated to a real device if one of the correct type and address is found on the network)

To delete a project Remote Device:

1. Select the project Remote Device by clicking the left hand button, the row will highlight
2. Press Delete Device on the Remote Device toolbar
3. The Remote Device will be removed from the project and, if no longer associated at all,the real device will move to the Unused list

Remote Device firmware

IMPORTANT: Remote Device firmware may need to be updated if a new version of Designer software has been installed. Devices with incompatible firmware will be highlighted in red.

To update a Remote Device's firmware:

1. Select the incompatible device by pressing the left hand button, the row will be highlighted
2. Press Reload Firmware on the Remote Device toolbar
3. The firmware update will proceed - you must not disturb this process

Associating Remote Devices

Unlike Controllers, which are uniquely associated with a project via their serial number, Remote Devices are associated by their address as selected on the unit itself. Fifteen automatic addresses (1>15) are provided with a manual option (M) for selecting more (16>100). Remote devices may share the same address and thus identity, useful for repeating a user interface at both ends of a corridor for example.

To associate a project Remote Device with a real device (automatic addresses 1>15):

1. Select the project Remote Device by clicking the left hand button, the row will highlight
2. Ensure that the device type and address matches a suitable unit, addressed at this number, found on the network
3. The real device will move from the Unused list and fuse with the project device so completing the row details

To associate a project Remote Device with a real device (manual addresses 16>100):

1. Ensure that the Remote Device is addressed to the "M" setting, you will need to note it's serial number (label on back)
2. Select the project Remote Device by clicking the left hand button, the row will highlight
3. Select the correct device type and the desired address in the range 16>100
4. Select the Remote Device's serial number from the pull-down menu of devices found on the network
5. The real device will move from the Unused list and fuse with the project device so completing the row details

Once all your project Remote Devices have been associated with real devices you can configure them, test your programming on the installation itself and finally upload to the Controllers for stand-alone operation.

Remote Input Output (RIO) device properties

Serial Port

The RIO 80, RIO 44 and RIO 08 have a multi-protocol serial port that can be configured to either RS232 full-duplex or RS485 half-duplex operation. The configuration options are:

- Type - select RS232 or RS485 as required
- Baud rate - select the baud rate
- Data bits - select the number of data bits (typically 8)
- Stop bits - select the number of stop bits
- Parity - select the parity type

I/O Configuration

The RIO 80, RIO 44 and RIO 08 differ by virtue of the number and type of I/O ports:

RIO 80	Eight inputs, no outputs & serial port
RIO 44	Four inputs, four outputs & serial port
RIO 08	No inputs, eight outputs & serial port

Inputs can be individually configured as either Contact Closure, Digital or Analog with the latter two modes allowing for the threshold or range to be selected. Outputs can be individually configured with a Startup state, whether the relay is on or off at startup.

See basic triggers for usage.

Audio

The stereo balanced line level audio input of a RIO A can be used for Audio triggers. Select the Audio button to enable this mode and to see the following configuration options:

- Route To - select the Audio Bus to route the incoming audio to
- Freq. Bands - select the number of frequency bands with which to analyse the incoming audio (max 30 per channel; the frequency bands sit along a logarithmic scale and have been chosen for an optimum response to music)
- Gain - turn Auto gain on or off, and set the manual gain level
- Peak Decay Rate - set the rate at which peaks in each frequency band will decay (the peak level can be used in triggers)
- Initially Enabled - the audio feed from a RIO A can be turned on or off by triggers, and here you can set the initial state

See advanced triggers for usage.

Timecode

The stereo balanced line level audio input of a RIO A can be used for timecode input. Select the Timecode button to enable this mode and to see the following configuration options:

- Channel - the audio input of the RIO A that the timecode input will be connected to
- Route To - select the Timecode Bus to route the timecode to

- Regenerate for - select the number of frames that will be generated by the RIO A's software flywheel in the event of a drop in timecode signal
- Ignore jumps for - select the maximum size of jump in incoming frames that will be ignored

MIDI

The RIO A has a MIDI input and output interface. This can either be used in Remote Device MIDI triggers, or it can receive MIDI timecode. The configuration options here are for MIDI timecode:

- Route To - select the Timecode Bus to route the MIDI timecode to
- Regenerate for - select the number of frames that will be generated by the RIO A's software flywheel in the event of a drop in MIDI timecode signal
- Ignore jumps for - select the maximum size of jump in incoming frames that will be ignored

DALI

The RIO D has a DALI bus interface. This can be used to control DALI ballasts via timeline programming or to receive DALI commands for use in DALI Input triggers.

Button Panel Station (BPS) device properties

Properties

The global properties for each BPS are set here:

- Minimum LED Intensity - set a percentage value as required, useful for ensuring that the buttons are always visible
- Held Timeout - set the amount of time in milliseconds that a button must be pressed to be considered as being held
- Repeat Interval - set the interval in milliseconds that a held button will transmit a repeat signal

Button Configuration

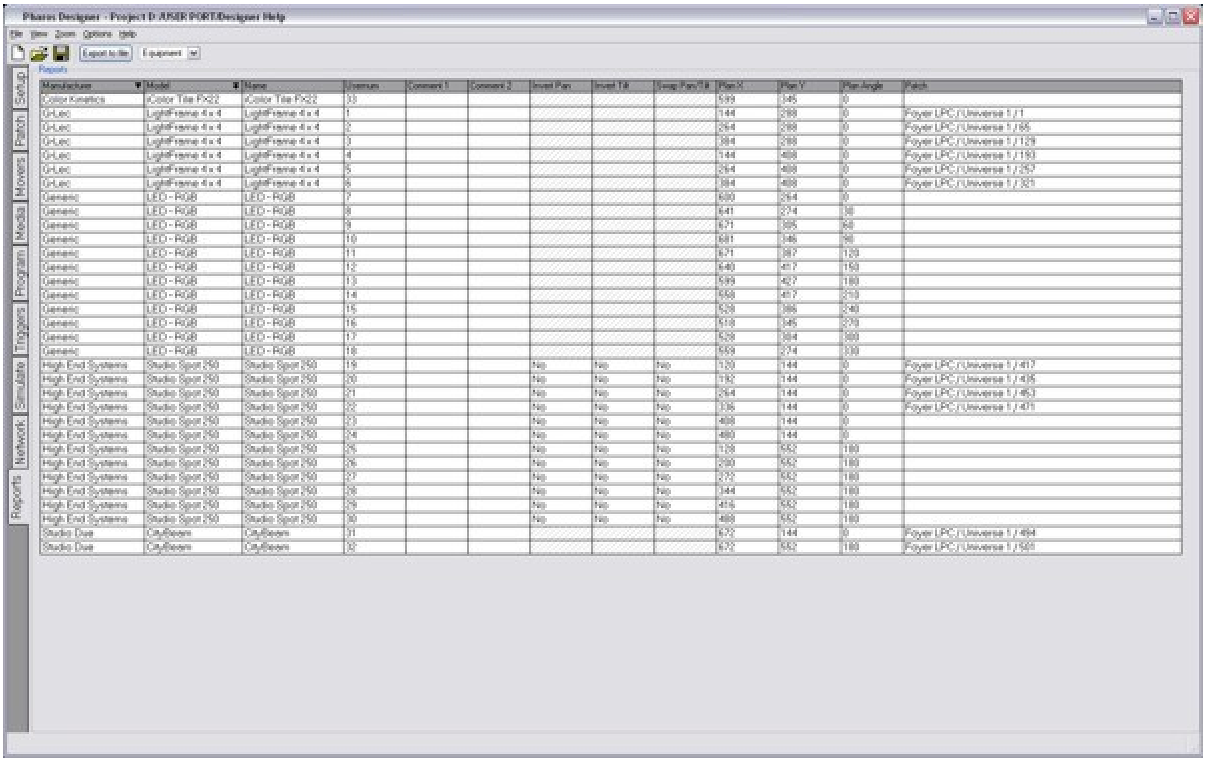
Each BPS has eight buttons with an integral white LED and the default setting for each button is set here:

- Effect - select the default LED effect (Off, Static, Slow Flash etc.)
- Intensity - set the default LED intensity (0% will equal the Minimum LED Intensity as set above)

See advanced triggers for usage and BPS learning IR for infrared operation.

Report

Designer can automatically produce reports to aid in producing documentation for the project:



Manufacturer	Model	Name	Usernum	Comment 1	Comment 2	Invert Pan	Invert Tilt	Swap Pan/Tilt	Plan X	Plan Y	Plan Angle	Patch
Color Kinetics	iColor Tile FX22	iColor Tile FX22	33						599	345	0	
G-Lec	LightFrame 4 x 4	LightFrame 4 x 4	1						144	288	0	Foyer LPC / Universe 1 / 1
G-Lec	LightFrame 4 x 4	LightFrame 4 x 4	2						264	288	0	Foyer LPC / Universe 1 / 65
G-Lec	LightFrame 4 x 4	LightFrame 4 x 4	3						384	288	0	Foyer LPC / Universe 1 / 129
G-Lec	LightFrame 4 x 4	LightFrame 4 x 4	4						144	408	0	Foyer LPC / Universe 1 / 193
G-Lec	LightFrame 4 x 4	LightFrame 4 x 4	5						264	408	0	Foyer LPC / Universe 1 / 257
G-Lec	LightFrame 4 x 4	LightFrame 4 x 4	6						384	408	0	Foyer LPC / Universe 1 / 321
Generic	LED - RGB	LED - RGB	7						600	264	0	
Generic	LED - RGB	LED - RGB	8						641	274	30	
Generic	LED - RGB	LED - RGB	9						671	305	60	
Generic	LED - RGB	LED - RGB	10						681	346	90	
Generic	LED - RGB	LED - RGB	11						671	387	120	
Generic	LED - RGB	LED - RGB	12						640	417	150	
Generic	LED - RGB	LED - RGB	13						599	427	180	
Generic	LED - RGB	LED - RGB	14						558	417	210	
Generic	LED - RGB	LED - RGB	15						528	386	240	
Generic	LED - RGB	LED - RGB	16						518	345	270	
Generic	LED - RGB	LED - RGB	17						528	304	300	
Generic	LED - RGB	LED - RGB	18						559	274	330	
High End Systems	Studio Spot 250	Studio Spot 250	19			No	No	No	120	144	0	Foyer LPC / Universe 1 / 417
High End Systems	Studio Spot 250	Studio Spot 250	20			No	No	No	192	144	0	Foyer LPC / Universe 1 / 435
High End Systems	Studio Spot 250	Studio Spot 250	21			No	No	No	264	144	0	Foyer LPC / Universe 1 / 453
High End Systems	Studio Spot 250	Studio Spot 250	22			No	No	No	336	144	0	Foyer LPC / Universe 1 / 471
High End Systems	Studio Spot 250	Studio Spot 250	23			No	No	No	408	144	0	
High End Systems	Studio Spot 250	Studio Spot 250	24			No	No	No	480	144	0	
High End Systems	Studio Spot 250	Studio Spot 250	25			No	No	No	128	552	180	
High End Systems	Studio Spot 250	Studio Spot 250	26			No	No	No	200	552	180	
High End Systems	Studio Spot 250	Studio Spot 250	27			No	No	No	272	552	180	
High End Systems	Studio Spot 250	Studio Spot 250	28			No	No	No	344	552	180	
High End Systems	Studio Spot 250	Studio Spot 250	29			No	No	No	416	552	180	
High End Systems	Studio Spot 250	Studio Spot 250	30			No	No	No	488	552	180	
Studio Due	CityBeam	CityBeam	31						672	144	0	Foyer LPC / Universe 1 / 494
Studio Due	CityBeam	CityBeam	32						672	552	180	Foyer LPC / Universe 1 / 501

Report toolbar

Use this to select one from the range of reports:

Equipment

Lists all the fixtures in the project. The complete fixture identification is shown complete with user annotation, plan position and patch information.

Group

Lists all the groups in the project. Members are listed and group usage is indicated.

Patch

Lists the complete patch data.

All Timelines

Provides a summary of all the timelines in the project.

Timeline

Provides a summary of each timeline. Use the pull-down to select the timeline.

Trigger

Provides a summary of the trigger programming. Complete with user annotation.

Network

Lists all the Controllers and Remote Devices in the project.

KiNet

Lists all the KiNet power supplies that have been added to Controllers in the project.

Font

Lists the fonts used in Dynamic Text presets on timelines in the project.

Report spreadsheet

All the reports are presented in spreadsheet form although none of the cells is editable; the reports just present an accurate account of the project programming.

The reports can be sorted and reorganised. Right-click on the column headings to set/clear primary/secondary sorts. Drag column headers to move them, drag the header divider lines to resize them. These spreadsheet layout settings are stored with the project.

Exporting a report

Press the "Export to file" button to save out a *.tsv file (Tab Separated Values) which can then be opened and formatted in a spreadsheet program ready for presentation.

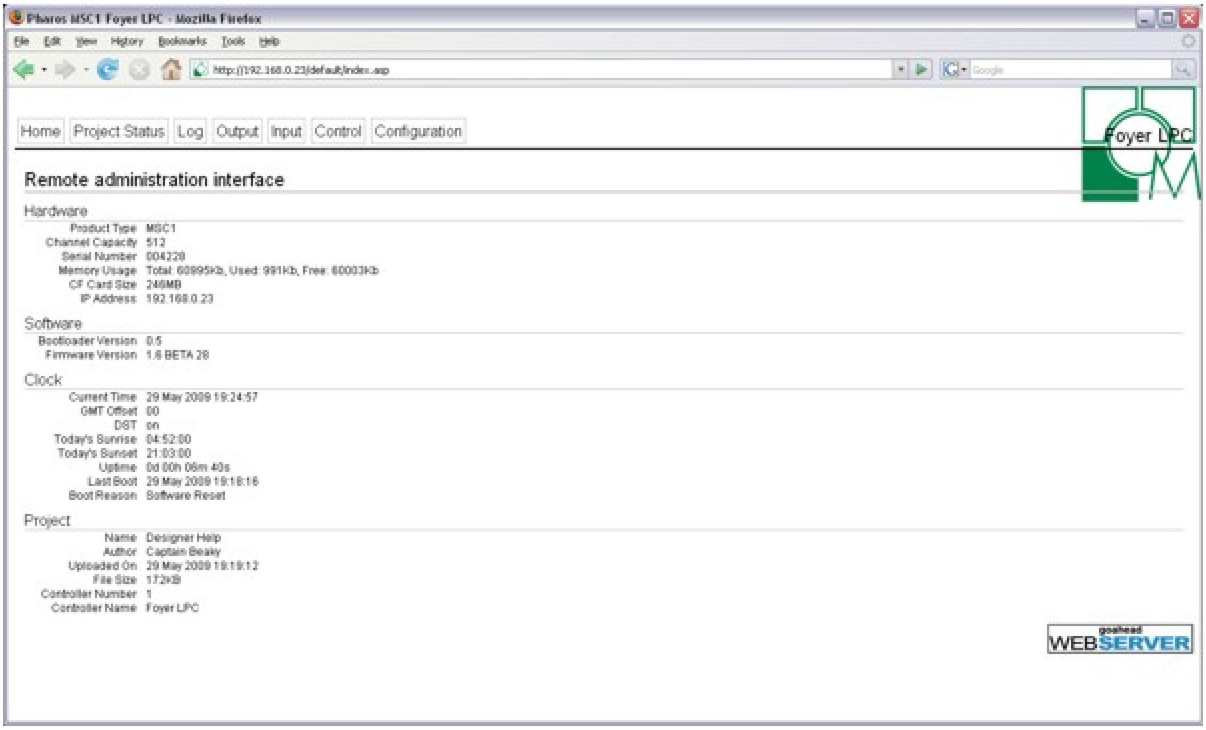
Web interface - Default pages

The Controller's internal web interface is a very powerful diagnostic and management tool. You can view a Controller's web interface from within Designer or, for remote access, browse to the index page at http://xxx.xxx.xxx.xxx/default/index.asp, where xxx.xxx.xxx.xxx is the IP address of the Controller.

Ethernet is the preferred connection to access the Controller's web interface. A USB connection can be used for configuration and troubleshooting provided that Designer is running but the Controller's playback performance may be affected.

Use the navigation buttons across the top to select these pages:

Home


Pharos MSC1 Foyer LPC - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://192.168.0.23/default/index.asp
Google
Home
Project Status
Log
Output
Input
Control
Configuration
Foyer LPC
goahead WEBSERVER

Remote administration interface

Hardware

Product Type	MSC1
Channel Capacity	512
Serial Number	004220
Memory Usage	Total: 60995Kb, Used: 991Kb, Free: 60003Kb
CF Card Size	246MB
IP Address	192.168.0.23

Software

Bootloader Version	0.5
Firmware Version	1.6 BETA 28

Clock

Current Time	29 May 2009 19:24:57
GMT Offset	00
DST	on
Today's Sunrise	04:52:00
Today's Sunset	21:03:00
Uptime	0d 00h 06m 40s
Last Boot	29 May 2009 19:18:16
Boot Reason	Software Reset

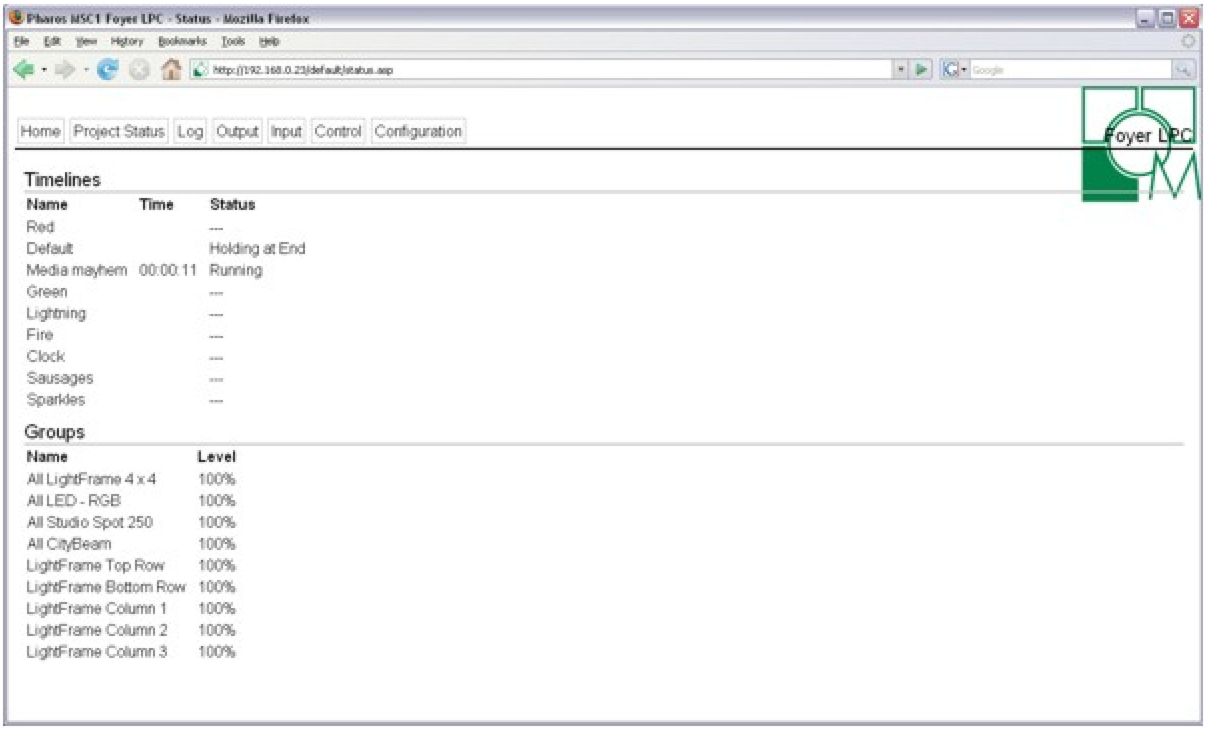
Project

Name	Designer Help
Author	Captain Beaky
Uploaded On	29 May 2009 19:19:12
File Size	172kB
Controller Number	1
Controller Name	Foyer LPC

The home page provides general information about the status of the Controller: Serial number, type, IP address, loaded project details and memory usage. The bootloader and firmware versions are also given.

For the MSC X (only) particular attention should be paid to the temperature readings; insufficient ventilation may cause the ambient temperature to rise and thus system & CPU temperatures to reach excessive levels, degrading performance.

Project Status


Pharos MSC1 Foyer LPC - Status - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://192.168.0.23/default/status.asp
Home
Project Status
Log
Output
Input
Control
Configuration
Foyer LPC

Timelines

Name	Time	Status
Red		---
Default		Holding at End
Media mayhem	00:00:11	Running
Green		---
Lightning		---
Fire		---
Clock		---
Sausages		---
Sparkles		---

Groups

Name	Level
All LightFrame 4 x 4	100%
All LED - RGB	100%
All Studio Spot 250	100%
All CityBeam	100%
LightFrame Top Row	100%
LightFrame Bottom Row	100%
LightFrame Column 1	100%
LightFrame Column 2	100%
LightFrame Column 3	100%

The status page provides feedback on the current state of playback:

Timelines

All timelines are listed with their current state and running time:

---	Inactive, the timeline has not run since the last reset
Running	The timeline is running and contributing to the output (items "on stage")
Running (Inactive)	The timeline is running in the background and not contributing to the output, generally because it has been overridden
Halted	The timeline is halted and contributing to the output
Halted (Inactive)	The timeline is halted in the background and not contributing to the output
Holding at End	The timeline is holding at end and contributing to the output
Holding at End (Inactive)	The timeline is holding at end in the background and not contributing to the output
Released	Inactive, the timeline has run but has been explicitly released

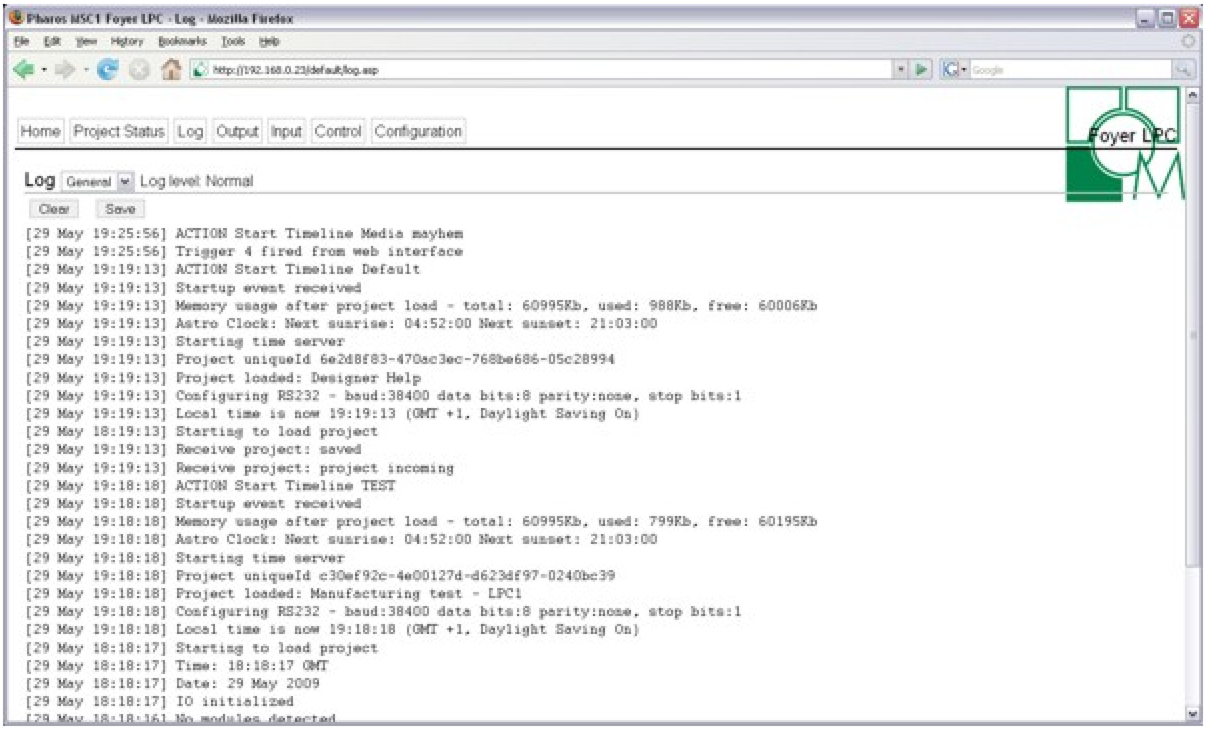
In systems with more than one Controller it is important to understand that this timeline status is only pertinent to the Controller being accessed. For example, the accessed Controller may report that a timeline is Running (Inactive) because its fixtures are not contributing to the output while another Controller may well be Running (Active) because its fixtures are contributing to the output. In such systems the complete status can only be determined by interrogating all Controllers.

Groups

All groups are listed with their current intensity level.

Use this page in conjunction with the Control and Log pages to interrogate and debug an installation.

Log



Pharos MSC1 Foyer LPC - Log - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://192.168.0.23/default/log.asp
Home
Project Status
Log
Output
Input
Control
Configuration
Foyer LPC
Log
General
Clear
Save

Log level: Normal

```
[29 May 19:25:56] ACTION Start Timeline Media mayhem
[29 May 19:25:56] Trigger 4 fired from web interface
[29 May 19:19:13] ACTION Start Timeline Default
[29 May 19:19:13] Startup event received
[29 May 19:19:13] Memory usage after project load - total: 60995Kb, used: 988Kb, free: 60006Kb
[29 May 19:19:13] Astro Clock: Next sunrise: 04:52:00 Next sunset: 21:03:00
[29 May 19:19:13] Starting time server
[29 May 19:19:13] Project uniqueId 6e2d8f83-470ac3ec-768be686-05c28994
[29 May 19:19:13] Project loaded: Designer Help
[29 May 19:19:13] Configuring RS232 - baud:38400 data bits:8 parity:none, stop bits:1
[29 May 19:19:13] Local time is now 19:19:13 (GMT +1, Daylight Saving On)
[29 May 18:19:13] Starting to load project
[29 May 19:19:13] Receive project: saved
[29 May 19:19:13] Receive project: project incoming
[29 May 19:18:18] ACTION Start Timeline TEST
[29 May 19:18:18] Startup event received
[29 May 19:18:18] Memory usage after project load - total: 60995Kb, used: 799Kb, free: 60195Kb
[29 May 19:18:18] Astro Clock: Next sunrise: 04:52:00 Next sunset: 21:03:00
[29 May 19:18:18] Starting time server
[29 May 19:18:18] Project uniqueId c30ef92c-4e00127d-d623df97-0240bc39
[29 May 19:18:18] Project loaded: Manufacturing test - LPC1
[29 May 19:18:18] Configuring RS232 - baud:38400 data bits:8 parity:none, stop bits:1
[29 May 19:18:18] Local time is now 19:18:18 (GMT +1, Daylight Saving On)
[29 May 18:18:17] Starting to load project
[29 May 18:18:17] Time: 18:18:17 GMT
[29 May 18:18:17] Date: 29 May 2009
[29 May 18:18:17] IO initialized
[29 May 18:18:16] No modules detected
```

The log can be cleared and saved to file using the Clear and Save buttons. Two types of log are provided:

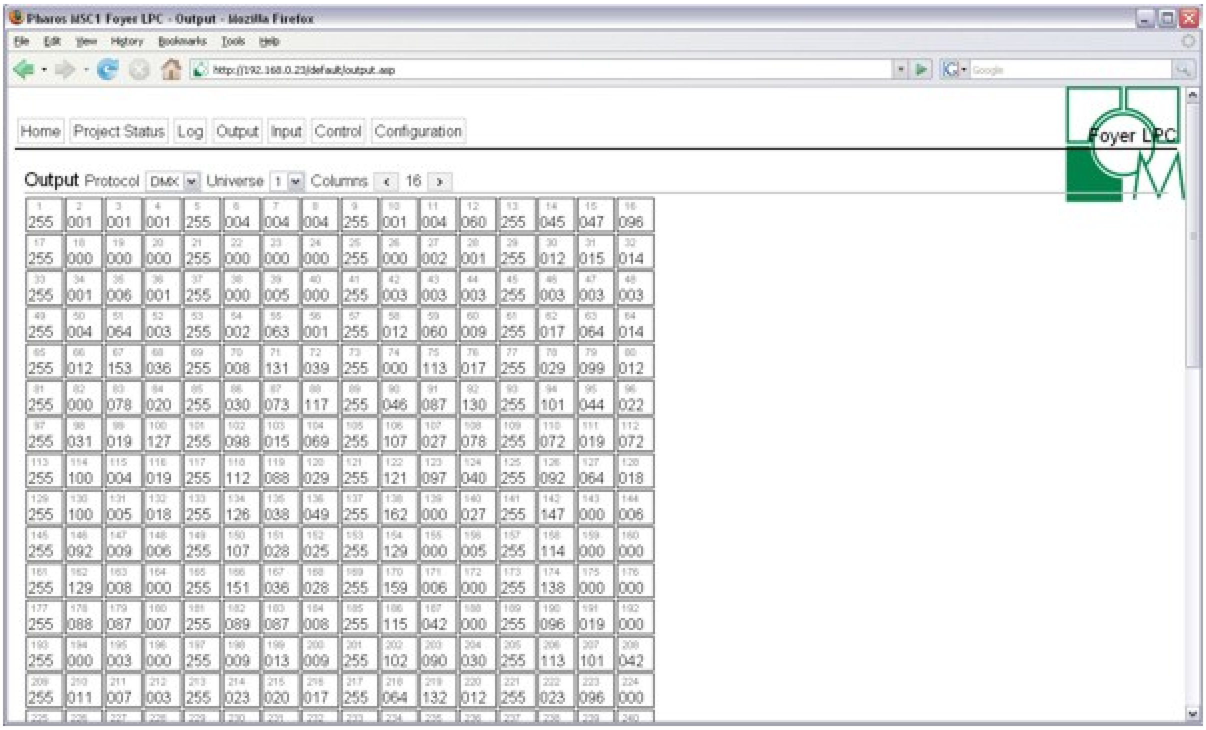
General log

A blow-by-blow account of all activity including input/output,RS232 serial strings for example, and trigger matching. Extremely useful in helping debug complex interfacing and triggering arrangements. Alternatively, the log can be viewed directly from within Designer over an Ethernet or USB connection using View > Controller Log.

System log

A less verbose log of the Controller's system activity, useful for examining the boot-up sequence to help debug problems.

Output


Pharos MSC1 Foyer LPC - Output - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://192.168.0.23/default/output.asp
Home
Project Status
Log
Output
Input
Control
Configuration
Foyer LPC

Output Protocol DMX Universe 1 Columns ‹ 16 ›

1 255	2 001	3 001	4 001	5 255	6 004	7 004	8 004	9 255	10 001	11 004	12 060	13 255	14 045	15 047	16 096
17 255	18 000	19 000	20 000	21 255	22 000	23 000	24 000	25 255	26 000	27 002	28 001	29 255	30 012	31 015	32 014
33 255	34 001	35 006	36 001	37 255	38 000	39 005	40 000	41 255	42 003	43 003	44 003	45 255	46 003	47 003	48 003
49 255	50 004	51 064	52 003	53 255	54 002	55 063	56 001	57 255	58 012	59 060	60 009	61 255	62 017	63 064	64 014
65 255	66 012	67 153	68 036	69 255	70 008	71 131	72 039	73 255	74 000	75 113	76 017	77 255	78 029	79 099	80 012
81 255	82 000	83 078	84 020	85 255	86 030	87 073	88 117	89 255	90 046	91 087	92 130	93 255	94 101	95 044	96 022
97 255	98 031	99 019	100 127	101 255	102 098	103 015	104 069	105 255	106 107	107 027	108 078	109 255	110 072	111 019	112 072
113 255	114 100	115 004	116 019	117 255	118 112	119 088	120 029	121 255	122 121	123 097	124 040	125 255	126 092	127 064	128 018
129 255	130 100	131 005	132 018	133 255	134 126	135 038	136 049	137 255	138 162	139 000	140 027	141 255	142 147	143 000	144 006
145 255	146 092	147 009	148 006	149 255	150 107	151 028	152 025	153 255	154 129	155 000	156 005	157 255	158 114	159 000	160 000
161 255	162 129	163 008	164 000	165 255	166 151	167 036	168 028	169 255	170 159	171 006	172 000	173 255	174 138	175 000	176 000
177 255	178 088	179 087	180 007	181 255	182 089	183 087	184 008	185 255	186 115	187 042	188 000	189 255	190 096	191 019	192 000
193 255	194 000	195 003	196 000	197 255	198 009	199 013	200 009	201 255	202 102	203 090	204 030	205 255	206 113	207 101	208 042
209 255	210 011	211 007	212 003	213 255	214 023	215 020	216 017	217 255	218 064	219 132	220 012	221 255	222 023	223 096	224 000
225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240

View output

Select the Protocol/DMX Port to examine a numerical snapshot of the control data being output, refreshed every 5 seconds. Select DVI to examine a graphical snapshot of the pixel matrix output.

Use in conjunction with Control and Status pages to debug an installation.

Park and Unpark

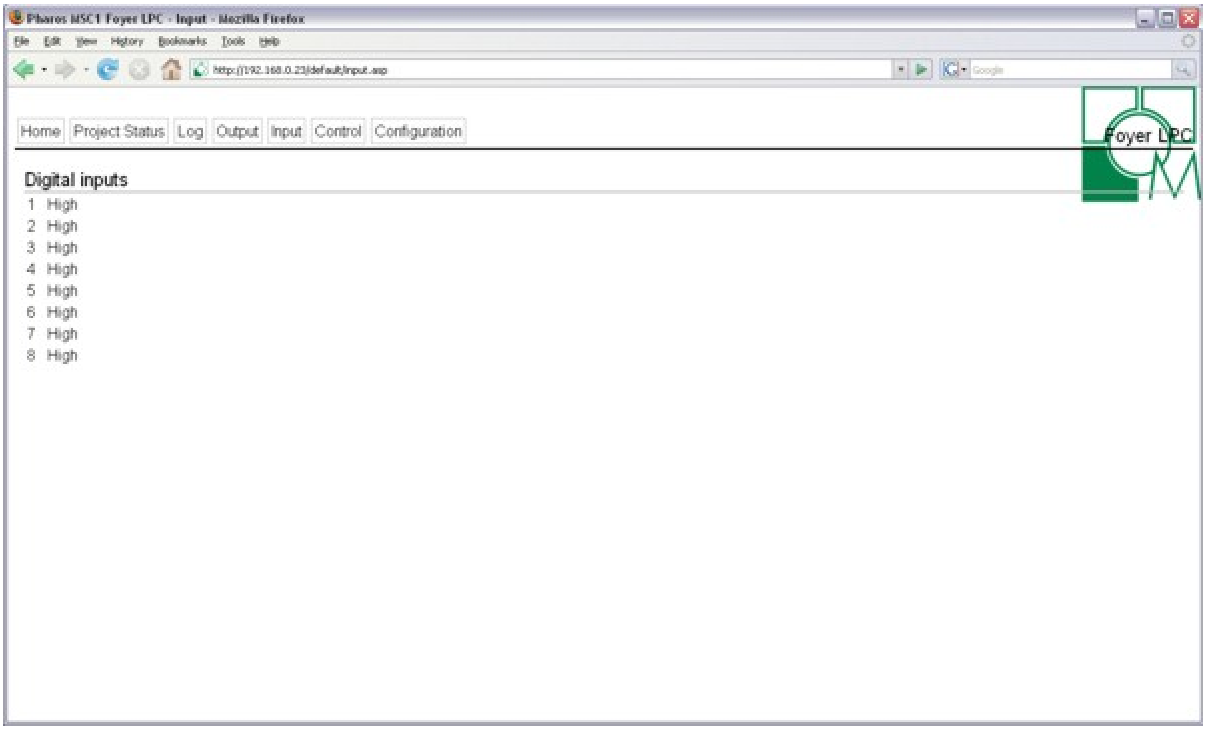
Password protected if set. Enter "admin" for the User Name and then the password.

Park allows you to lock the value of a particular channel without actually altering your programming. This can be useful to turn off a fixture that is misbehaving temporarily or to make sure a working light stays on while you are programming.

Park can be accessed from the output view of the web interface, simply enter the channel or range of channels and the value at which to park. Parked channels are shown in red within the output view. There is the option to Unpark from the same view.

Parked channels will remain parked when you upload shows or output live. However all parked channels will be cleared if the Controller is reset or the power is cycled.

Input



Digital inputs

1	High
2	High
3	High
4	High
5	High
6	High
7	High
8	High

DMX

Select the DMX input or Expansion Module(s) (if any connected) to examine a snapshot of the DMX values being input, useful for debugging DMX triggering and control.

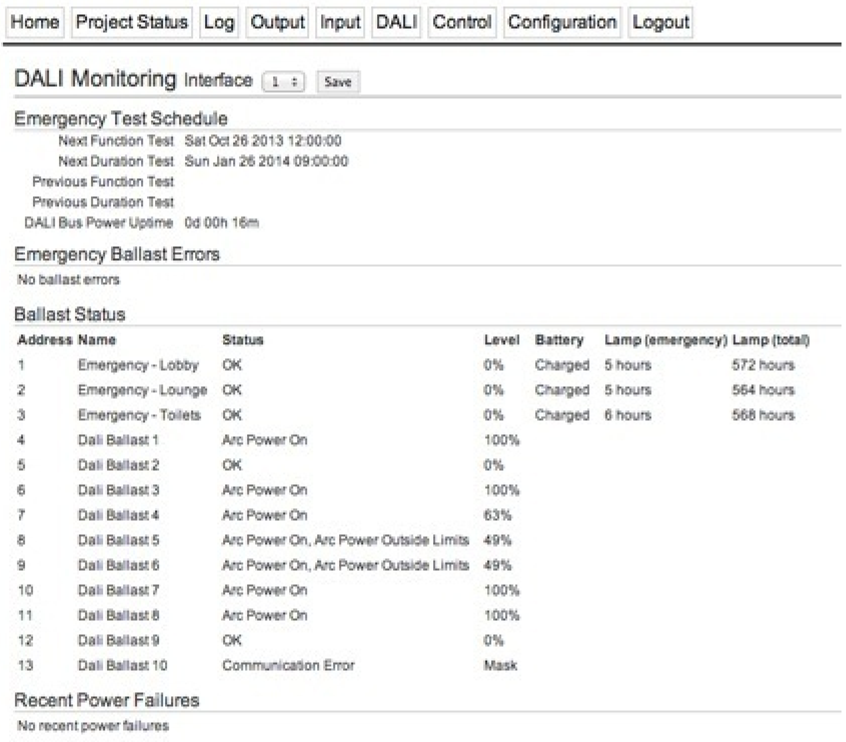
Digital inputs

Use to examine the status of the Controller's digital inputs.

DALI

CSV Export

This information can be viewed in a comma separated values file by clicking the Save button. Copy, paste and save the information you require in a separate document.



DALI Monitoring Interface 1 Save

Emergency Test Schedule

Next Function Test	Sat Oct 26 2013 12:00:00
Next Duration Test	Sun Jan 26 2014 09:00:00
Previous Function Test	
Previous Duration Test	
DALI Bus Power Uptime	0d 00h 16m

Emergency Ballast Errors

No ballast errors

Ballast Status

Address	Name	Status	Level	Battery	Lamp (emergency)	Lamp (total)
1	Emergency - Lobby	OK	0%	Charged	5 hours	572 hours
2	Emergency - Lounge	OK	0%	Charged	5 hours	564 hours
3	Emergency - Toilets	OK	0%	Charged	6 hours	568 hours
4	Dali Ballast 1	Arc Power On	100%			
5	Dali Ballast 2	OK	0%			
6	Dali Ballast 3	Arc Power On	100%			
7	Dali Ballast 4	Arc Power On	63%			
8	Dali Ballast 5	Arc Power On, Arc Power Outside Limits	49%			
9	Dali Ballast 6	Arc Power On, Arc Power Outside Limits	49%			
10	Dali Ballast 7	Arc Power On	100%			
11	Dali Ballast 8	Arc Power On	100%			
12	Dali Ballast 9	OK	0%			
13	Dali Ballast 10	Communication Error	Mask			

Recent Power Failures

No recent power failures

Emergency Test Schedule

Only populated if the current interface has emergency ballasts present. View information about when Emergency Ballast tests are due to take place as well as the time and date of previous tests. Also view the uptime of the DALI bus.

Emergency Ballast Errors

Lists all reported errors reported by emergency fixtures on the current DALI interface. Errors will show ballast address, tests failed and reported errors.

Once a ballast has been repaired it can be marked as fixed here or by using triggers. Once a ballast has been marked as fixed it will remain in the Ballast Errors section until a subsequent test has confirmed that the fixture is indeed operational again.

Ballast Status

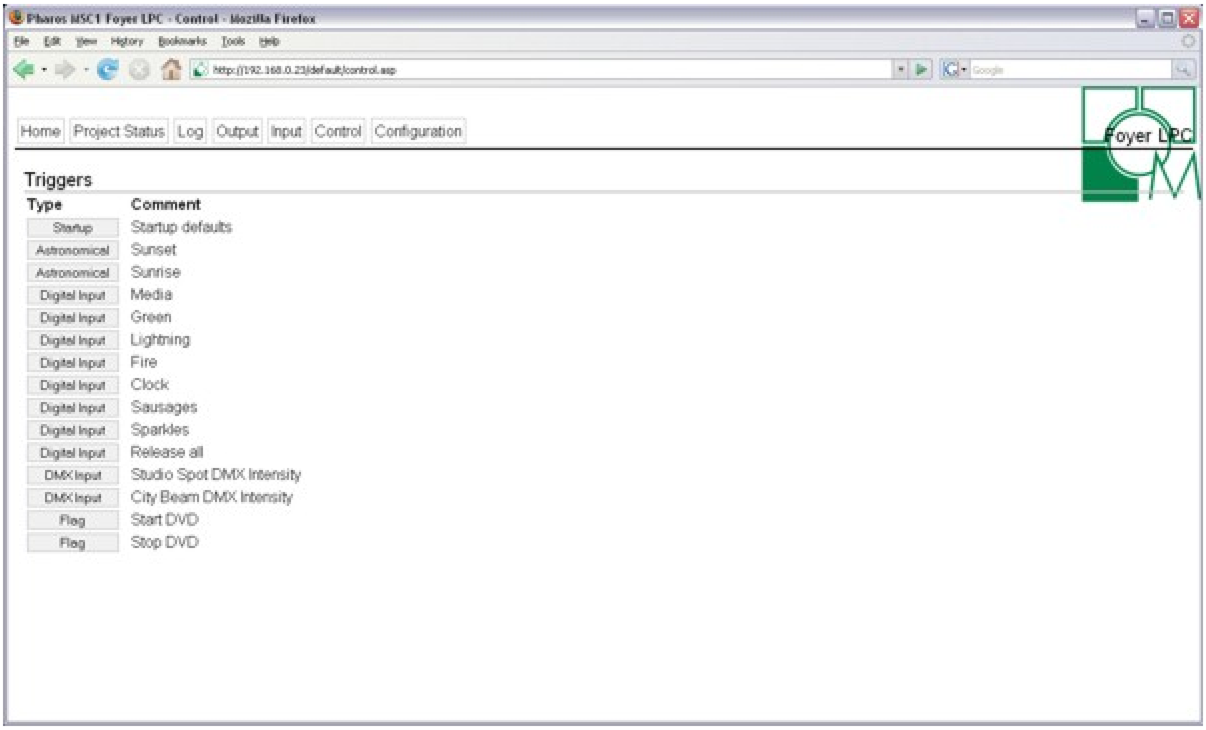
Lists all ballasts and reported status on the current interface. Standard ballasts show the ballast Address, Name, Status and Level; emergency ballasts also show Battery Charge, Emergency Lamp Hours and Total Lamp Hours.

Recent Power Failures

Lists any reports of bus power failures for the current interface.

Control

Password protected if set. Enter "admin" for the User Name and then the password.


Pharos MSC1 Foyer LPC - Control - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://192.168.0.23/default/control.asp
Google
Home
Project Status
Log
Output
Input
Control
Configuration
Foyer LPC

Triggers

Type	Comment
Startup	Startup defaults
Astronomical	Sunset
Astronomical	Sunrise
Digital Input	Media
Digital Input	Green
Digital Input	Lightning
Digital Input	Fire
Digital Input	Clock
Digital Input	Sausages
Digital Input	Sparkles
Digital Input	Release all
DMX Input	Studio Spot DMX Intensity
DMX Input	City Beam DMX Intensity
Flag	Start DVD
Flag	Stop DVD

A Controller can be controlled remotely in two ways:

Command line

An advanced feature that allows direct control of a specific Controller's fixtures, timelines and even DMX channels via the script engine, see command line reference.

Triggers

All the triggers in the project, together with user annotation, are listed here and can be fired by clicking on them. Since a network of multiple Controllers share triggers, firing triggers from one Controller's web interface will trigger all the Controllers in the project.

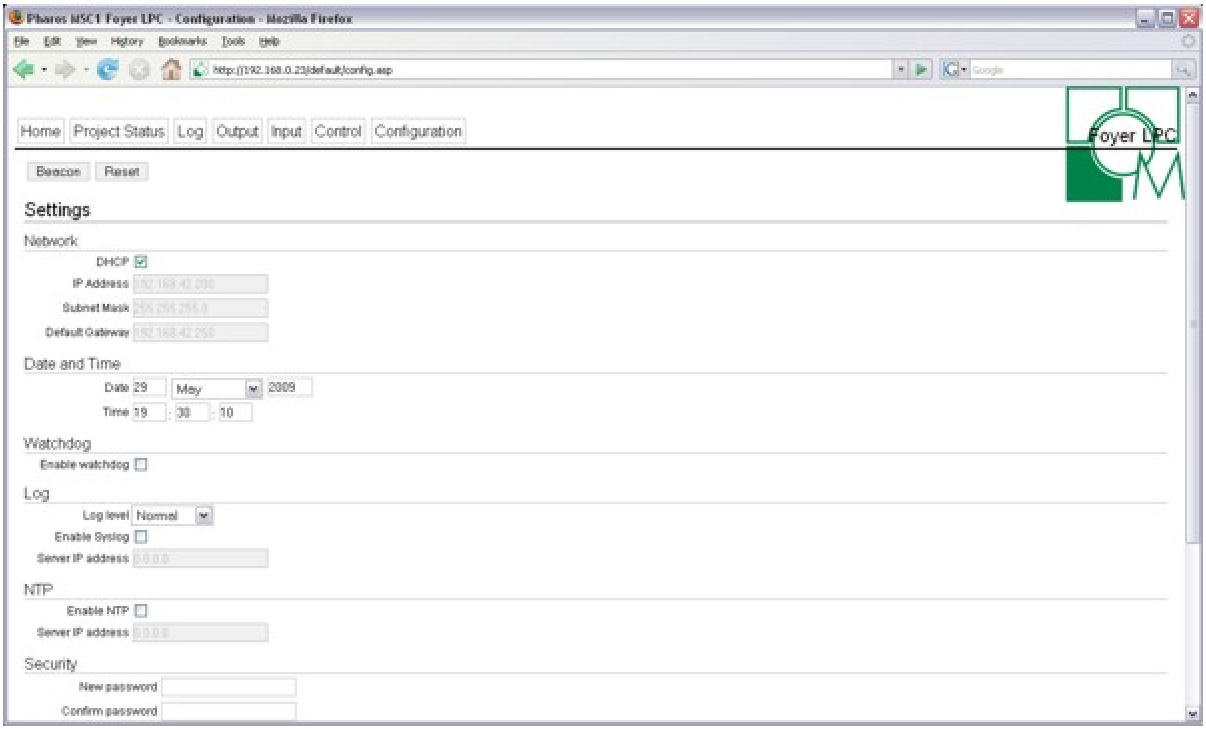
Triggers on the default Control web page will not test conditions, create a custom page if condition testing is required.

Dynamic text slots

All the Dynamic text slots are listed with their current value. You can edit any text slot and changes will take effect according to the preset settings on the timeline (immediately, next cycle, on timeline restart).

Configuration

Password protected if set. Enter "admin" for the User Name and then the password.


Pharos MSC1 Foyer LPC - Configuration - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://192.168.0.23/default/config.asp
Google
Home
Project Status
Log
Output
Input
Control
Configuration
Foyer LPC
Beacon
Reset

Settings

Network

DHCP ☑

IP Address 192.168.42.200

Subnet Mask 255.255.255.0

Default Gateway 192.168.42.250

Date and Time

Date 29 May 2009

Time 19 : 30 : 10

Watchdog

Enable watchdog ☐

Log

Log level Normal

Enable Syslog ☐

Server IP address 0.0.0.0

NTP

Enable NTP ☐

Server IP address 0.0.0.0

Security

New password

Confirm password

All the Controller's configuration settings are displayed and can be changed here, see configuration for details.

Remote upload

In addition, at the bottom of the page, is the means to upload a project file remotely via the web interface as an alternative to uploading directly from Designer. See the network section to learn how to generate a file for remote uploading.

IMPORTANT: Controllers must be running the same version of firmware as the Designer software. Uploading a project file to a Controller running different firmware may result in the project failing to load and run. Check the Controller's home page to determine compatibility before attempting a remote upload.

Note that the Safari browser is unable to request progress information while uploading a project file. This feature is not currently supported on Google's Chrome browser. As an alternative, use the Firefox browser.

Custom Page

If a default custom page is specified, a Custom Page button will appear beside Configuration.

Web interface - Command line

The Controller has a command line entry box in the Control page of its web interface. Text entered in this command line is interpreted by a Lua script specified by the user as part of the project configuration. Users may write their own scripts if they wish (see Lua scripts) but a standard script "commandline.lua" is provided in the resources directory of Unison Mosaic Designer.

IMPORTANT: Note that by default there is no command line script installed. Most installations will not require a command line and so it is inactive by default. If you wish to use the standard command line script you must use the "Custom Command Line Parser" option on the File menu to select the script.

The command line syntax defined in the standard commandline.lua script has the following commands, where x,y and z represent numbers and [] indicates optional syntax:

Selections

x
x-y
x/y
x-y/z

where x, y and z are the fixture number, '-' selects a range and '/' combines discrete selections or ranges.

Setting intensity

x@y[%][tz]

where x is the fixture number, y is the level (either as a DMX value or as a percentage) and z is an optional time in seconds. If a time is not specified then it is treated as a snap change.

Examples:

1@127	Set intensity of fixture 1 to 127 immediately
2@50%	Set intensity of fixture 2 to 50% (127) immediately
3@100%t5.5	Set intensity of fixture 3 to 100% fading over 5.5 seconds

Setting RGB

Setting red, green and blue uses the same syntax as intensity, but replacing the @ with r for red, g for green, and b for blue.

Examples:

1r255	Set red of fixture 1 to 255 immediately
3g0	Set green of fixture 3 to 0 immediately
7b100%t2	Set blue of fixture 7 to 100% fading over 2 seconds

Note that the default values for red, green and blue are 100% (255) to give white. So to make a fixture output the colour red then you will need to set green and blue to zero.

You can also apply multiple settings to the same selection of fixtures in a single command. For example:

1-25@100%r255b255g0	Set fixtures 1 thru 25 to 100% intensity, red to 255, blue to 255 and green to 0 immediately

Clearing fixture settings

xc[ty]

where x is the fixture number and y is an optional time in seconds.

Examples:

1c	Clear settings for fixture 1 immediately
5ct6.5	Clear settings for fixture 5 fading over 6.5 seconds

Clearing all fixtures settings

ca[tx]

where x is an optional time in seconds.

Examples:

ca	Clear settings for all fixtures immediately
cat10	Clear settings for all fixtures fading over 10 seconds

Multiple Commands

Multiple commands can be applied from a single command line if separated by commas.

Examples:

1@100%,1r0,1b0,1g255	Set intensity of fixture 1 to 100%, red and blue to 0 and green to 255
1ct5,4r255,4@75%t5	Clear settings for fixture 1 fading over 5 seconds, set red for fixture 4 to 255 immediately and then set intensity of fixture 4 to 75% fading over 5 seconds

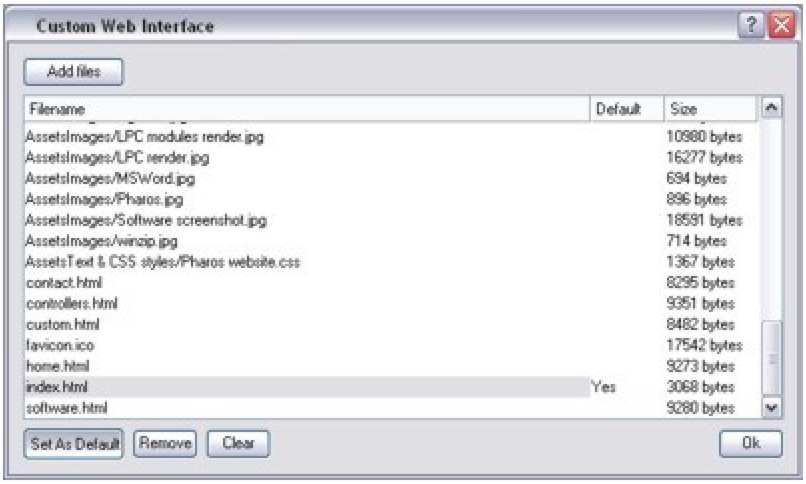
Interaction with timeline playback

Settings applied from the command line are applied as if from a high priority timeline, so they will override all normal timeline programming until cleared. Fades to and from command line settings behave just like fades between timelines.

Web interface - Custom page(s)

To add a custom web page, or set of pages, to the web interface on a Controller, go to the File menu and choose Custom Web Interface.

This will open a dialog that shows the files that currently make up your custom web interface:



Filename	Default	Size
AssetsImages/LPC modules render.jpg		10980 bytes
AssetsImages/LPC render.jpg		16277 bytes
AssetsImages/MSWord.jpg		694 bytes
AssetsImages/Pharos.jpg		896 bytes
AssetsImages/Software screenshot.jpg		18591 bytes
AssetsImages/winzip.jpg		714 bytes
AssetsText & CSS styles/Pharos website.css		1367 bytes
contact.html		8295 bytes
controllers.html		9351 bytes
custom.html		8482 bytes
favicon.ico		17542 bytes
home.html		9273 bytes
index.html	Yes	3068 bytes
software.html		9280 bytes

Adding files

To add files to the web interface, click Add Files. You will be prompted for a directory that contains the files to add. The contents of the chosen directory will be added to the root of the web interface. If any filenames clash, they will be overwritten. Sub-directories will be recursively added, with the exception of directories called "default", "query" or "trigger", which are reserved names.

Any file type can be included, but since they are stored as part of the project file, be aware that they will take away from space available for programming, and will increase show upload times.

Removing files

To remove a file, select the file and press Remove.

To remove all files, press Clear.

Defining the default page

To pick a default file, select the file and press Set As Default. The default page will be displayed when you browse to http://xxx.xxx.xxx.xxx, where xxx.xxx.xxx.xx is the IP address of the Controller.

Defining triggers

Custom web pages can trigger the Controller by creating a hyperlink to "/trigger/xyz", where "xyz" is the trigger number, displayed in the first column of the Triggers management window. Clicking on this hyperlink will fire the numbered trigger, if it exists, but will not cause the page to refresh so there is no need to use Java Script tricks to prevent the page from flickering.

Injecting variables

You can capture variables and inject them into the numbered trigger by specifying a "var" field in the query string. The value of this field is expecting a comma-separated list of values, with each value in the format "abc:def" where "abc" is the captured value and "def" is the range of the value (optional). If the value of "abc" is not a number, it is treated as 0. If value of "def" is not a number, it is treated as 255.

For example, the URL "/trigger/1?var=14" will fire trigger number 1 and will inject the value 14. If this trigger had a Start Timeline action that expected a variable to select the timeline, then it would start timeline 14.

In another example, the URL "/trigger/2?var=50:100" will fire trigger number 2 and will inject the value 50. If the variable is used by an action that is expecting a value within a range, for example Set intensity expecting a level between 'off' and 'full', then the injected variable will be treated as 50% since 50:100 = 50%.

You can also inject multiple variables. For example, "/trigger/2?var=50:100,3,4:16" will inject 50 (50%), 3 and 4 (25%).

If you want to inject a string to a variable, for example if you want to set the contents of a text slot, the characters must be bound by double quotes (" or %22 if using character escaping). To include a " in the injected string you must prefix it with a backslash. To include a backslash, write two backslashes. Here are some examples for injecting strings:

- /trigger/1?var="hello" -> (hello)
- /trigger/1?var=%22hello%22 -> (hello)
- /trigger/1?var="hello \"world\"" -> (hello "world")
- /trigger/1?var="hello\\world" -> (hello\world)

Anything after the closing " and before the next comma is ignored:

- /trigger/1?var="hell"o,"world" -> (hell),(world)

If a closing quote is missing, the remaining string is used:

- /trigger/1?var=1,"hello,2 -> (1),(hello,2)

Testing conditions

By default, when firing a trigger from a custom web interface, conditions on that trigger will be tested.

To disable this, specify "conditions=0" in the URL query string. For example, "/trigger/1?conditions=0" will fire trigger 1 regardless of whether its conditions are satisfied.

Triggers on the default Control web page will not test conditions.

Dynamic feedback

To provide feedback on a custom web page you can make use of the query interface to access timeline status information, text slot contents, etc.

Issues

When you Upload, Designer will check things like triggers and hardware configuration to make sure that there are no inconsistencies. If any issues are found, the Issues tab will be opened automatically and a description of each issue will be listed so that you can take corrective action (or you can ignore and proceed if you like):

Issue - '?' will provide the specific details:	Solution:
Controller '?' isn't associated with a physical controller	Associate the Controller, see Controller association
Controller '?' is associated with serial number '?', which isn't found on the network	Connect the Controller, see Controller connection
Controller '?' has an IP address clash	Check the Controller's IP settings don't clash with any other devices on the network, see Controller configuration
Controller '?' is found on the network but has the wrong firmware	Update the Controller's firmware, see Controller connection
Controller '?' is on the wrong network	Change the Controller's IP settings, see Controller configuration
Controller '?' uses the same network for management and protocol ports	MSC X network interfaces must be connected to the same physical network if they are both on the same IP network. This issue serves as a reminder.
The time server cannot support all of the RIO Ds in the project	There are more RIO Ds in the project than the Controller set as the time server can support, see DALI.
Controller '?' new location information will make internal clock differ from local time	The PC running Designer and the Controller are in different time zones, see Location
Controller '?' internal clock differs from local time by greater than 5 minutes	Synchronize the Controller's realtime clock to the PC, see date and time
Controller '?' has Reset to Defaults DIP switch set	Clear the DIP switch setting (for factory testing use only)
Controller '?' has disable hardware watchdog DIP switch set	Clear the DIP switch setting (for factory testing use only)
Controller '?' has serial debug DIP switch set	Clear the DIP switch setting (for factory testing use only)
Controller '?' has configuration for a '?' module that is not attached to physical controller '?'	Attach the Expansion Module
Controller '?' doesn't have configuration for a '?' module that is attached to physical controller '?'	Module attached to the wrong Controller, move it or reconfigure
Controller '?' has no DMX fixtures patched to it	Patch the fixtures (or ignore if working with DALI only)

DALI programming has changed on Interface '?' since the ballast configuration was last uploaded	Update the DALI ballasts, see DALI
Timeline '?' uses Timecode Bus '?' but no input is connected to that source	Ensure that the timecode source routed to that bus
Timeline '?' uses Audio Bus '?' but no input is connected to that source	Ensure that the audio source routed to that bus
Timeline '?' uses band '?' on Audio Bus '?' which is not provided by the assigned input	Select a band that is available for that bus, or change the source for the bus
Trigger '?' will not be matched because controller '?' is not a TPC	Reconfigure the trigger with the correct Controller number
Trigger '?' is set to match on controller '?', but no controller has that number	Reconfigure the trigger with the correct Controller number
'?' is set to match with '?' module number '?' on any controller, but no such module is configured	Add and configure the Expansion Module
'?' is set to match with '?' module number '?' on controller '?', but no such module is configured	Add and configure the Expansion Module
'?' is set to match with '?', but no such device is configured	Add and configure the Remote Device
'?' has an incomplete remote device identifier	Reconfigure the trigger or action with the correct Remote Device identifier
There are Astronomical triggers in the project, but the project location appears to have not been set	Set a location for the project, see Location
Trigger '?' requires input '?' on controller '?' to be configured as a digital input or contact closure	The Controller's input has been configured incorrectly, see Controller Interfaces
Trigger '?' requires input '?' on controller '?' to be configured as an analog input	The Controller's input has been configured incorrectly, see Controller Interfaces
Trigger '?' is set to match with controller '?', which doesn't have digital inputs	Choose a Controller that does have digital inputs
Trigger '?' is set to match with controller '?', which doesn't have analog inputs	Choose a Controller that does have analog inputs
Trigger '?' requires a serial port, but controller '?' doesn't have a serial port	Choose a Controller that does have a serial port
Trigger '?' is set to match with controller '?', which requires the serial port to be set to 1	Controller only supports one serial port, so set the trigger serial port to 1
Trigger '?' requires the serial port on controller '?' to be set to RS232	Reconfigure the serial port of the Controller to use the correct mode, see Controller Interfaces
Trigger '?' uses the RS485 serial port, but controller '?' doesn't have an RS485 serial port	Reconfigure the trigger to use a Controller that does have an RS285 serial port

Trigger '?' requires a valid RS485 port number	Reconfigure the trigger with the correct port number (1, unless using two RS485 modules on one Controller, in which case 2 may be valid)
Trigger '?' requires the serial port on controller '?' to be set to RS485	Reconfigure the serial port of the Controller to use the correct mode, see Controller Interfaces
Trigger '?' requires a valid DMX In port number	Reconfigure the trigger with the correct port number (1, unless using two DMX-In modules on one Controller, in which case 2 may be valid)
Trigger '?' is set to match with controller '?', which requires the DMX In port to be set to 1	Controller only supports one DMX input port, so set the trigger DMX port to 1
Trigger '?' requires the serial port on controller '?' to be set to DMX	Reconfigure the serial port of the Controller to use the correct mode, see Controller Interfaces
Trigger '?' is set to match with controller '?', but that controller has disabled DMX In	Reconfigure the DMX input for the Controller, see Controller Interfaces.
Trigger '?' uses the MIDI In port, but controller '?' doesn't have a MIDI In port	Reconfigure the trigger to use a Controller with a MIDI In port
Trigger '?' requires a valid Audio module number	Reconfigure the trigger with the correct Module number
Trigger '?' requires a valid DALI Slave module number	Reconfigure the trigger with the correct Module number
Condition '?' will not be tested on controller '?' because the controller doesn't support digital inputs	Reconfigure the trigger with a controller that does support digital inputs
Condition '?' requires input '?' on controller '?' to be configured as a digital input or contact closure	The Controller's input has been configured incorrectly, see Controller Interfaces
Condition '?' will not be tested on controller '?' because the controller doesn't support analog inputs	Choose a Controller that does have analog inputs
Condition '?' on trigger '?' requires a valid DMX In port number	Reconfigure the trigger with the correct port number
Action '?' is set to run on controller '?', but no controller has that number	Reconfigure the trigger with the correct Controller number
Action '?' will not be run because controller '?' isn't a TPC	Reconfigure the trigger with the correct Controller number
Action '?' requires a serial port, but controller '?' doesn't have a serial port	Choose a Controller that does have a serial port
Action '?' is set to use controller '?', which requires the serial port to be set to 1	Controller only supports one serial port, so set the action serial port to 1
Action '?' requires the serial port on controller '?' to be set to RS232	Reconfigure the serial port of the Controller to use the correct mode, see Controller Interfaces

Action '?' uses the RS485 serial port, but controller '?' doesn't have an RS485 serial port	Reconfigure the action to use a Controller that does have an RS285 serial port
Action '?' on trigger '?' requires a valid RS485 port number	Reconfigure the trigger with the correct port number
Action '?' requires the serial port on controller '?' to be set to RS485	Reconfigure the serial port of the Controller to use the correct mode, see Controller Interfaces
Action '?' uses the MIDI Out port, but controller '?' doesn't have a MIDI Out port	Reconfigure the trigger to use a Controller with a MIDI Out port
Action '?' is using the same variable to select the BPS number and the button number	Check the variables assigned to these fields in the action properties
Trigger '?' requires input '?' on '?' to be configured as a digital input or contact closure	The RIO's input has been configured incorrectly, see Remote Devices
Trigger '?' requires '?' to be configured as an Audio input device	The RIO has been configured incorrectly, see Remote Devices
Trigger '?' uses frequency band '?' on '?' but that device doesn't provide that band	Select a band that is available for that device, or reconfigure the RIO, see Remote Devices
Condition '?' requires input '?' on '?' to be configured as a digital input or contact closure	The RIO's input has been configured incorrectly, see Remote Devices
Condition '?' expects all inputs on '?' to be configured as either digital inputs or contact closures	The RIO's input has been configured incorrectly, see Remote Devices
Condition '?' requires input '?' on '?' to be configured as an analog input	The RIO's input has been configured incorrectly, see Remote Devices
Ethernet Bus '?' uses port '?'. This clashes with the internal web server and will prevent correct operation of controller '?'	Reconfigure the Ethernet bus to use a different port
Ethernet Bus '?' uses port '?'. This clashes with file transfer and will prevent correct operation of controller '?'	Reconfigure the Ethernet bus to use a different port
Controller '?' is outputting KiNet so ensure the DHCP server on protocol network interface is serving an IP address on the same network as the patched power supplies	Configure the DHCP server to use a 10.xxx.xxx.xxx IP address
Controller '?' is outputting KiNET so ensure that the protocol network interface IP address is on the same network as the patched power supplies	Reconfigure the Protocol port to use a 10.xxx.xxx.xxx IP address
Multiple KiNet power supplies have the IP address '?'	Reconfigure power supplies to use unique IP addresses
Controller '?' is outputting KiNet so ensure that the controller IP address is on the same network as the patched power supplies	Reconfigure the Controller to use a 10.xxx.xxx.xxx IP address

Controller '?' doesn't have a valid IP address for Network 2	Reconfigure the Protocol port to use a valid IP address
Controller '?' doesn't have a interface file for the screen	The TPC in the project doesn't have an associated Touch Editor file, see Controller Properties
Controller '?' (TPC) has been configured with a TPC-RIO but there is no TPC-RIO discovered	Connect the Controller to a TPC-RIO or uncheck the 'Configure TPC-RIO' checkbox in Controller Interfaces.
Controller '?' (TPC) has patch for DMX 1 but has not been configured with a TPC-RIO or proxy MSC	Connect the Controller to a TPC-RIO or MSC 1 for DMX output. Alternatively remove DMX patch from the controller.

Frequently asked questions

Software

Is the free software a cut-down demo version?

No. The free Designer software is the full software package. Downloads and updates can be found at our website.

Does the Designer software support the Apple Macintosh?

Yes, from v1.2 we have full Mac OS X support and from v1.2.4 universal binary support for both PowerPC and Intel Macs. Project files will be compatible, regardless of the operating system on which they are created.

What are the PC minimum requirements for Designer software?

- Windows XP 32bit (service pack 2 or later), Windows Vista 32bit and 64bit, Windows 7 32bit and 64bit
- Apple Quicktime (6.5 or later)
- Intel Pentium or PowerPC processor at 1 GHz or above
- 256MB RAM
- 100MB free hard-disk space
- 1024x768 minimum screen resolution (higher is better)
- OpenGL graphics acceleration
- Ethernet or USB connection

Are project files compatible across versions and platforms?

Any project file saved in an earlier version of Designer can be loaded by a later version. Project files are compatible between the PC and Mac versions of the software.However, a project file saved in a later version of Designer may not be backward compatible as we reserve the right to make structural changes to improve the product.

Can I have multiple versions of Designer on my computer?

No, instead keep the installers for each version readily available. A reinstall takes just seconds and updates the desktop shortcut.

What documentation is available?

For setup and configuration of the hardware we provide an Installation Guide. This is available as a PDF and a printed version is shipped with every Controller. There is user help available within the software (this document), this is also available as a PDF for printing. Visit our website for more resources.

How many timelines can I program? How many fixtures, etc?

See system limits and capacities.

How can I tell what DMX levels are being generated?

During programming, when simulating using Output Live, there is a DMX viewer available in the View menu which displays the DMX values generated by Designer. During Controller playback you can use the web interface to view the Controller's DMX output.

Where's Undo?

Sorry, there isn't one (yet). Remember that accidental drags can be aborted by hitting Esc. For more serious mishaps, Designer can make backup copies of the project every time you save so you can always revert to this last saved version, see below.

Backing up?

Designer can keep a number old versions of the project file when you save. In the Preferences dialog you can set the number of old files to keep. Before saving your project, Designer will rename the project file on disk by adding the current time and date to the file name, such as my_project_bak_2007-04-18_15-58-09.mdp. If you already have the number of specified backups, the oldest backup will be removed from the disk.

The rest is up to you so save early, save often. Use File > Save As to produce manual backups of the project at each important programming milestone.

What are the Unison Mosaic Designer file extensions?

*.mdp	Unison Mosaic Designer Project file
*.mde	Unison Mosaic Designer Exported project file, contains referenced media & background plan image so can grow quite large, use this to transfer and archive projects
*.wrn	MSC project data file, uploaded to the MSC's memory card or saved to disc for remote web interface upload
*.phv	Custom video format used by the AVC
*.pha	Custom audio format used by the AVC
*.avc	AVC project data file, uploaded to the AVC's memory card or saved to disc for remote FTP upload with the media files
*.dat	AVC font file
*.mdx	Generating when exporting media slots for media manager
*.mtc	User interface file for the Tessera Panel Controller

Can the project file be retrieved from the Controller(s)?

No, not unless you explicitly store it,as an exported project (*.mde) file, on the Controller's memory card. To do this you must remove the card from the Controller and plug it into the PC or use the File Transfer feature.

How best to archive a project?

Export the project (File > Export or Ctrl + E) and save the *.mde file on one or more of the Controllers and keep a version for yourself safe, together with the relevant Designer installer. If you lose the project file you will have to start from scratch.

How do I programme RS232, RS485 or Ethernet triggers?

Unison Mosaic Controllers can send and receive triggers from 3rd party devices over RS232 Serial, RS485 Serial and Ethernet protocols. Programming these all involve entering a string of characters in the trigger parameter pane. These can be entered in three formats; ASCII, Hex & Decimal.

Essentially, the format and string entered depends entirely on the other device. In the Designer trigger/action fields enter exactly the same string as the other device sends, or is expecting to receive. Simply match the string.

Some devices will have a fixed string. For example, the Dynalite button panel sends "1c0164000000ff<c>" if button 1 is pressed. Therefore in an RS485 trigger the same hex characters are entered in the String box.

Other devices may accept customisable messages, allowing meaningful names, descriptions or comments to be entered in an ASCII format (ASCII being standard letters & numbers). In this instance you may create a string in the other device such as "play timeline 6". Then in Designer, the ASCII string entered for the trigger will also be "play timeline 6". The action will be to start timeline 6. For a useful user interface it may be that the user choice and string may be descriptive, such as "red walls, blue ceiling". The trigger string in Designer will be "red walls, blue ceiling", and start the appropriate timeline without the user needing to know what the timeline number is for the desired look.

Variables can also be used within these Serial and Ethernet strings. A message such as "GOxx" is valid, where the xx is replaced with a two digit decimal number which will start the timeline with the corresponding number, rather than having to assign distinct strings to every timeline. In Designer the string would be "GO<2d>" which indicates the system will expect a 2 digit variable. The action would still be Start Timeline, but rather than selecting a specific timeline, set the Variable Index to 1.

A useful tip for programming the Controller is to be connected and utilize the Controller's log. Any messages on the line will appear here, be it in nice friendly ASCII, or other formats. The string can be copied straight out of the log and pasted into the parameters string box - eliminating the possibility for typos, etc. The trigger types can be mapped out in advance with comments and actions, then whatever the other device sends can be grabbed accurately at the time. The log is a great troubleshooting tool for checking what triggers have been received and what actions are fired as a result.

Fixtures

What happens if I need a fixture that isn't in the library?

If you're keen you can make your own fixture by grappling with the fixture editor. Preferably please contact support with your requirements.

I have a fixture with lots of DMX modes, which mode should I use?

The "flat" mode that addresses each cell/element/pixel individually with no additional intensity master nor effects/macro channels. Designer has been developed to get the most out of compound fixtures, typically LED battens, tubes and tiles, by driving each pixel individually, without "help" or complication - virtual intensity channels are created as required and arrays can be constructed onto which effects and media, far more powerful than any built-in function, can be applied and precisely controlled. If in doubt please contact support.

Hardware

Which revision of MSC 1 & 2 hardware do I have?

MSCs with serial numbers 006xxx and higher are a revised version of the original hardware. The core functionality is the same, but the revised hardware features improved performance and enhanced connectivity, such as RS485 and DMX inputs. Below is a list of the distinguishing features to help tell the hardware apart from a visual inspection:

- Product name on the front label of the revised MSC is **MSC**; product name on the original hardware is **MSC 1** or **MSC 2**. The channel capacity of the revised hardware is indicated on the serial number on the rear of the unit.
- Revised MSC has an Output LED on the front; original hardware does not.
- Revised MSC has a 3-pin multi-protocol serial connecter; original hardware has a 9-pin D-type RS232 connector.
- Revised MSC has no expansion bus connector on the side; original hardware does.
- Revised MSC has an SD card slot on the side; original hardware has a Compact Flash card slot on the side.

The additional features of the revised MSC are:

- Faster processor for improved performance.
- 8 multi-mode inputs: digital, contact closure and analog, configurable in Designer.
- Multi-protocol serial interface: RS232 or RS485, including DMX input.
- SD card for project file storage for greater compatibility with today's PCs and Macs.
- Output LED to indicate eDMX or raw DMX output.

What show control interfaces does the MSC 1, 2 & 4 support?

By providing RS232/RS485 serial (including DMX input), MIDI,Ethernet & digital/analog inputs, the MSC can interface with many generic, off-the-shelf products and devices. Via the built-in web interface any browser (PC, Mac, PDAs, etc) can utilise the hyperlinks for triggering and a custom web interface can be designed to provide a user-friendly skin. Remote Devices provide further interfacing options e.g. remote RS232/RS485, SMPTE/EBU timecode, audio input, DALI, MIDI, etc.

Is Unison Mosaic RDM compatible?

Yes, the hardware and Designer software supports RDM and we will continue to add features in future software versions.

Will I need more memory on the Controller?

The MSC 1, 2 & 4 ships with a 256MB memory card, the MSC X & AVC 4GB. Project data is very memory efficient and, in multi-Controller systems, each Controller only stores what it needs and so it is unlikely that you will need more memory.

However, heavy use of imported media may necessitate an upgrade (256MB stores 2 hours on a fully patched MSC 2) and so you can easily replace the card for one of greater capacity, please contact support for recommended cards.

Are there any diagnostic tools?

The LED status indicators on the Controllers serve a dual purpose. In normal operation they indicate system functionality and activity. In an error state, they provide diagnostics, refer to the Installation Guide for details. There is also an MSC recovery procedure and an MSC X Recovery Tool.

When should I use reset?

The reset button provides a convenient way to cycle power. It has exactly the same effect. There is no recommendation to reset the Controllers periodically.

Should I keep Controllers in the field up-to-date with the latest firmware?

No, not unless you know that a problem you are having would be solved with an update or you need to change the programming and need the new Designer features. For minor tweaks it's probably best to install the relevant Designer version and do it with that. As a rule, if it's working, leave it be.

What warranty does ETC offer?

Unison Mosaic hardware is warranted for 5 years. Please contact support if you are experiencing any issues.

What user serviceable parts are there in a Controller or Remote Device?

The Unison Mosaic product range has been designed for longevity and reliability. There is almost nothing user serviceable apart from a battery for the realtime clock and the DMX driver ICs. Please contact support if you are experiencing any issues.

Standards compliance?

The Unison Mosaic product range is manufactured to the highest quality in compliance with international standards, refer to the product's Installation Guide for details.

Network

What are the differences between connecting to a PC via USB and Ethernet?

The USB connection is all about temporary convenience, hence it’s location on the front of the box; the main connectors may be difficult to get to in a fully wired equipment rack. Out of the box, the initial computer connection is via USB, to set up IP an address in readiness for an Ethernet connection. It is an easy way to connect again locally to edit and upload a project but USB has some disadvantages:

- Output Live refresh rates are lower
- Limited cable length

How do the Unison Mosaic products cope with sharing a network with other, non-lighting devices?

Unison Mosaic products can happily sit on any network. They do not broadcast a high volume of messages and will only listen to Unison Mosaic specific messages.

What about remote focus units, portable control stations, IR, etc?

With a wireless network access point, any PDA with wireless capability can be set to browse to the Controller's web interface. The web interface includes status monitoring and logging, hyperlinks of all trigger events and a command line.

The BPS and TPC have learning IR sensors, so they can be programmed to respond to any IR remote.

Is there a way to call up channels for focus?

Yes, on the Control page of the web interface there is a command line that allows the user to enter fixture intensity & RGB values. This also provides a park function as values set here override any programmed values.

Troubleshooting

The following section lists common problems and their solutions, beginning with an explanation of the Controller's LED indicators:

What are the Controller's LEDs telling me?

MSC and AVC Status LEDs

The ETC logo will illuminate when power is applied to the Controller. The red LEDs on the top/front of the Controller indicate the unit’s current status:

- The Active LED illuminates once the boot-up procedure has completed and is indicative of a fully functional unit.
- The Ethernet LED(s) indicates network activity (not network link) while the remaining LEDs indicate communication on the various ports of the Controller.
- The DMX (original MSC 1 & 2), Output (TPC, MSC and MSC X) and PAL/NTSC (AVC) LEDs indicate that a valid project file has been loaded from the memory card and that playback has started.

TPC Status LEDs

The top red LED on the front and rear of the Controller will illuminate when power is applied. The other red LEDs on the front of the TPC (visible when the magnetic overlay is removed) indicate the unit's current status:

- The Active LED illuminates once the boot-up procedure has completed and is indicative of a fully functional unit.
- The Ethernet LED indicates network activity (not network link).
- The Output LED indicates that a show is loaded and eDMX data is being output.

Error codes

Additionally the red status LEDs are used to indicate any boot failures of the Controller that prevent the unit from going active. Refer to the Installation Guide for details.

Why can't I see the Controller in the Designer network window?

Presuming that the Controller has successfully booted its firmware (thus its Active LED illuminated) then there is a communication problem between the Controller and the PC running Designer:

USB problems

- Windows: Did the USB driver install when you connected the Controller for the first time? Try unplugging the USB cable,wait for a bit,then plug it into a different port, the Found New Hardware wizard should open to guide you through the process, this will happen twice.
- Is the USB cable too long or damaged? Are you using an extension cable? Are you using a hub? Try a direct connection with a short (1M) cable.
- Is the Controller correctly earthed and a connection made between chassis ground (earth) and signal ground (0V)?
- Is the Controller's firmware compatible with Designer? Update it with the MSC recovery procedure or MSC X Recovery Tool.

- Still not working? Have you installed any other software with USB functionality (especially PC lighting software with USB "DMX dongles") since installing Designer? Unplug the Controller and any 3rd party USB DMX device, uninstall and reinstall Designer, reboot the PC and try again.
- Have you installed an earlier version of Designer on top of a later version? If so, reinstall the latest version you've used, uninstall it, then reinstall the earlier version you want to use again. It is best practice to always uninstall Designer before installing another version to ensure any files and drivers are properly cleaned up.

Ethernet problems (network)

- Quit and restart Designer again once you're sure that the network is up, use your PC's LAN status tools.
- By default, the Controllers are set to obtain an IP address from a DHCP server, is there one on the network? Put one up and reset the Controller or set a static IP address via USB.
- Do you have more than one network connection on the PC? Wireless perhaps? If so, did you choose the right network, the one with the Controller(s), when you started Designer?
- Is the Controller's firmware compatible with Designer? Update it with the TPC recovery procedure, the MSC recovery procedure or MSC X Recovery Tool.
- Are there any managed switches on the network? Traffic storms from 3rd party devices? Try "pinging" the Controller and other network debugging ploys beyond the scope of this document.

Ethernet problems (one-to-one)

- For a direct, one-to-one connection between a computer and the AVC you must use a crossover cable. You can, however, use a normal cat-5 cable for the TPC, the revised MSC hardware and the MSC X because their network interfaces are auto-sensing.
- Quit and restart Designer again once you're sure that the is network up, use your PC's LAN status tools.
- By default, the Controllers are set to obtain an IP address from a DHCP server, is the PC running one? Set a static IP address for both parties, the Controller via USB.
- Do you have more than one network connection on the PC? Wireless perhaps? If so, did you choose the right network, the one with the Controller, when you started Designer?
- Is the Controller's firmware compatible with Designer? Update it with the TPC recovery procedure, the MSC recovery procedure or MSC X Recovery Tool.

Incorrect Ethernet cable (CAT5/5E/6) pairing

Not all electrical installers are aware of the subtleties of Ethernet cabling, in particular the correct pairing scheme. While incorrectly paired short cables may work, longer cables almost certainly won't or may exhibit intermittent errors. Note that simple continuity testers will NOT expose an incorrectly paired cable. See this Wikipedia topic for details.

I can see the Controller in the network window but it is shown in grey?

Controllers must be on the same Ethernet subnet as the PC running Designer. Select the controller and change it's IP settings accordingly.

I can see the Controller in the network window but it is shown in red?

Controllers must be running the same version of firmware as the Designer software. Controllers with incompatible firmware will be highlighted in red. Select the Controller in the network window and press Reload Firmware.

Simulation looks fine but when I upload to the MSC nothing happens?

- Fixtures not patched. Try Output Live or examine the DMX Viewer to debug.
- Output Live left turned on (although a dialog now warns of this when uploading).
- The MSC or TPC hasn't received a valid trigger to commence playback. Use the web interface to check status, examine the log and stimulate triggers.

Trigger conditions do not work in simulation, why?

Trigger conditions are not tested by the simulator.

Output Live does nothing?

- Fixtures not patched.
- Output Live has been disabled for one or more of the TPCs or MSCs.

Why do I see a delay between the simulation and the MSC in Output Live?

If you are connecting to the Controller via USB there may be a discernable delay. Use Ethernet instead, or upload the show to the Controller and run the show directly.

The Controller's playback performance is deteriorating over time, why?

If your project has large numbers of timelines set to Hold or Loop, and these timelines are never explicitly released, then over time they will build up in the background and cause the Controller to struggle. Program your triggers to ensure that such timelines are explicitly released when no longer needed.

Uploading was working ok but now always fails?

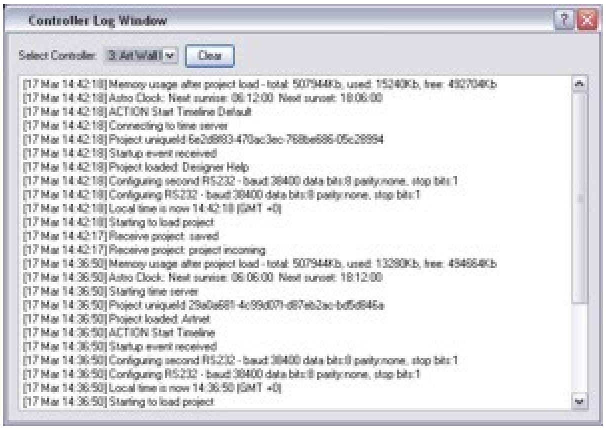
The memory card has become corrupt and must be formatted, use network configuration or the web interface.

When I try to Upload I see a list of issues instead?

Designer will check things like triggers and hardware configuration to make sure that there are no inconsistencies. If any issues are found, the Issues tab will be opened automatically and a description of each issue will be listed so that you can take corrective action, see Issues.

Is there a way of seeing what the Controller is doing?

Yes, Controllers generate a log which can be viewed either via the web interface or from within Designer using View > Controller Log on the main toolbar:



```
[17 Mar 14:42:18] Memory usage after project load - total: 507944Kb, used: 15240Kb, free: 492704Kb
[17 Mar 14:42:18] Astro Clock: Next sunrise: 06:12:00 Next sunset: 18:06:00
[17 Mar 14:42:18] ACTION Start Timeline Default
[17 Mar 14:42:18] Connecting to time server
[17 Mar 14:42:18] Project uniqueId 6e2d8f83-470ac3ec-768be686-05c28994
[17 Mar 14:42:18] Startup event received
[17 Mar 14:42:18] Project loaded: Designer Help
[17 Mar 14:42:18] Configuring second RS232 - baud:38400 data bits:8 parity:none, stop bits:1
[17 Mar 14:42:18] Configuring RS232 - baud:38400 data bits:8 parity:none, stop bits:1
[17 Mar 14:42:18] Local time is now 14:42:18 [GMT +0]
[17 Mar 14:42:18] Starting to load project
[17 Mar 14:42:17] Receive project: saved
[17 Mar 14:42:17] Receive project: project incoming
[17 Mar 14:36:50] Memory usage after project load - total: 507944Kb, used: 13280Kb, free: 494664Kb
[17 Mar 14:36:50] Astro Clock: Next sunrise: 06:06:00 Next sunset: 18:12:00
[17 Mar 14:36:50] Starting time server
[17 Mar 14:36:50] Project uniqueId 29a0a681-4c99d071-d87eb2ac-bd5d846a
[17 Mar 14:36:50] Project loaded: Artnet
[17 Mar 14:36:50] ACTION Start Timeline
[17 Mar 14:36:50] Startup event received
[17 Mar 14:36:50] Configuring second RS232 - baud:38400 data bits:8 parity:none, stop bits:1
[17 Mar 14:36:50] Configuring RS232 - baud:38400 data bits:8 parity:none, stop bits:1
[17 Mar 14:36:50] Local time is now 14:36:50 [GMT +0]
[17 Mar 14:36:50] Starting to load project
```

Use the Select Controller pull-down to view the Controller of interest, the verbosity (detail) of the log can be changed using Controller configuration.

I get a "Magic number does not match" error when opening a project file?

You are trying open a project file that was created and saved with a later major version of Designer, for example using Designer v1.1.x to open a file created with v1.2.x. While we endeavour to maintain backward compatibility with files saved with earlier versions, progress requires that we sometimes make architectural changes going forwards. Reinstall the later version.

I get loads of warnings about custom fixtures when opening a project file?

You are trying to open a project that contains custom fixtures with the same identification number as custom fixtures already on your PC. Either delete the custom fixtures on your PC or give them different identification numbers. This has been fixed in v1.2.3 so please contact support if you still see these warnings.

I have forgotten the Controller's password?

You will need to go on site and gain access to the Controller then contact support for further instructions.

I have checked the FAQ and troubleshooting but I'm still stuck?

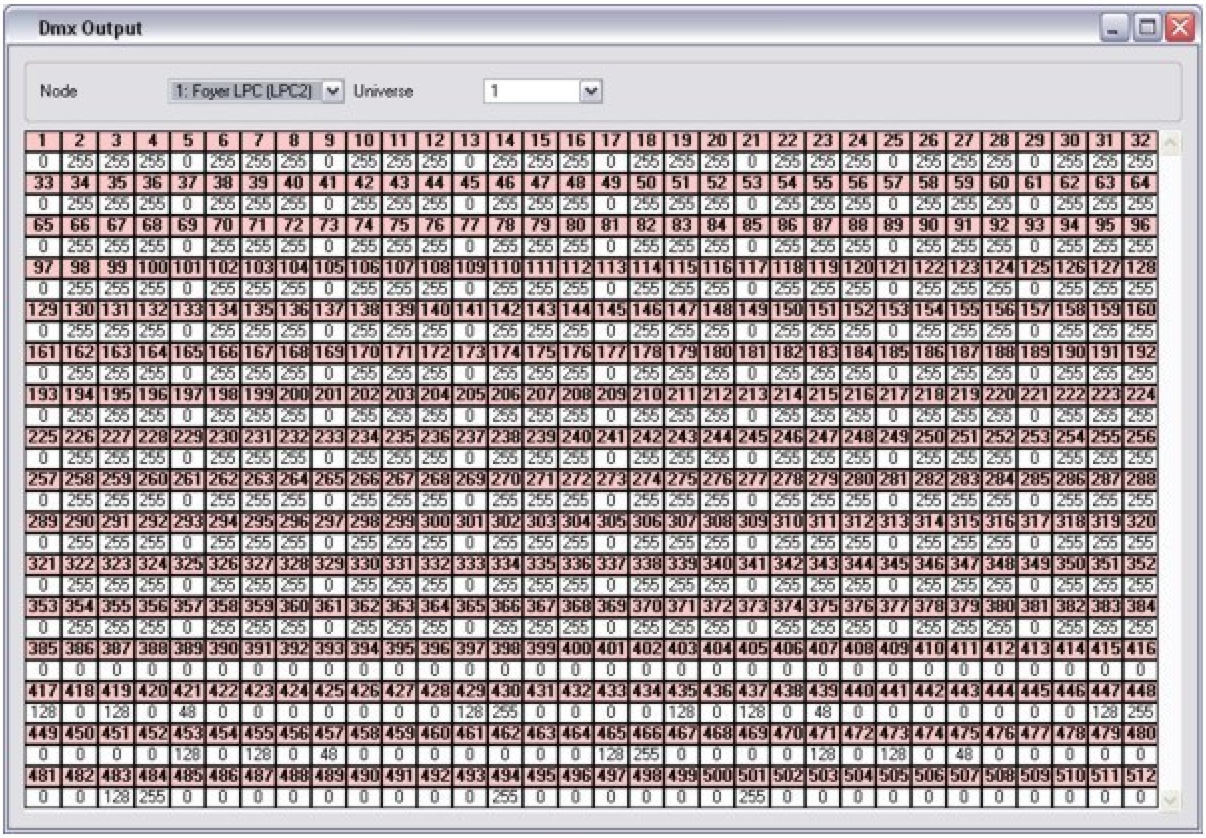
Contact support, please be prepared to send in your project file.

When using DMX In on a MSC or AVC, is my DMX line terminated?

No. To terminate the DMX line you should add a 120 Ohm resistor across the positive and negative terminals.

Output viewer

Select View > Output viewer on the main toolbar to open this window:



Node 1: Foyer LPC (LPC2) Universe 1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128
0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255
129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255
161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192
0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255
193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224
0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255
225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256
0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255
257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288
0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255
289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320
0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255
321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352
0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255
353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384
0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255	0	255	255	255
385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448
128	0	128	0	48	0	0	0	0	0	0	0	128	255	0	0	0	0	128	0	128	0	48	0	0	0	0	0	0	0	128	255
449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480
0	0	0	0	128	0	128	0	48	0	0	0	0	0	0	0	128	255	0	0	0	0	128	0	128	0	48	0	0	0	0	0
481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512
0	0	128	255	0	0	0	0	0	0	0	0	0	255	0	0	0	0	0	0	255	0	0	0	0	0	0	0	0	0	0	0

Use the Controller, Protocol and Universe pull-downs to select the Controller and DMX universe that you wish to view. What you will see depends on the status of your patch and the simulator:

Unpatched universe

If the universe is not patched then all values will be zero (0) regardless of the simulator's status.

Patched, simulator not running (reset)

If the universe is patched and the simulator is not running then you will see the default values for the fixtures. These are the values that the Controller will output after a reset or power cycle and prior to a timeline running, see Precedent.

Patched, simulator running (playing or paused)

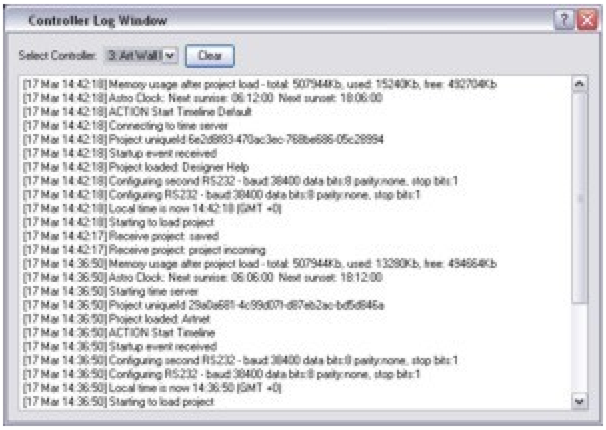
If the universe is patched and the simulator is running then you will see the values that the Controller will generate when it runs this timeline. Designer uses exactly the same playback algorithms as the Controller so what you see with the DMX viewer is what you'll get with the Controller.

Output Live

If you have connected Controllers then you will be able to select Output Live in the simulator to have Designer generate the DMX values directly, the Controllers acting purely as a DMX driver. In this case what you see in the DMX viewer is exactly what is being output to the fixtures in realtime.

Log viewer

Select View > Controller Log Window on the main toolbar to open this window:



```
[17 Mar 14:42:18] Memory usage after project load - total: 507944Kb, used: 15240Kb, free: 492704Kb
[17 Mar 14:42:18] Astro Clock: Next sunrise: 06:12:00 Next sunset: 18:06:00
[17 Mar 14:42:18] ACTION Start Timeline Default
[17 Mar 14:42:18] Connecting to time server
[17 Mar 14:42:18] Project uniqueId 6e2d8f83-470ac3ec-768be686-05c28994
[17 Mar 14:42:18] Startup event received
[17 Mar 14:42:18] Project loaded: Designer Help
[17 Mar 14:42:18] Configuring second RS232 - baud:38400 data bits:8 parity:none, stop bits:1
[17 Mar 14:42:18] Configuring RS232 - baud:38400 data bits:8 parity:none, stop bits:1
[17 Mar 14:42:18] Local time is now 14:42:18 (GMT +0)
[17 Mar 14:42:18] Starting to load project
[17 Mar 14:42:17] Receive project: saved
[17 Mar 14:42:17] Receive project: project incoming
[17 Mar 14:36:50] Memory usage after project load - total: 507944Kb, used: 13280Kb, free: 494664Kb
[17 Mar 14:36:50] Astro Clock: Next sunrise: 06:06:00 Next sunset: 18:12:00
[17 Mar 14:36:50] Starting time server
[17 Mar 14:36:50] Project uniqueId 29a0a681-4c99d07f-d87eb2ac-bd5d846a
[17 Mar 14:36:50] Project loaded: Artnet
[17 Mar 14:36:50] ACTION Start Timeline
[17 Mar 14:36:50] Startup event received
[17 Mar 14:36:50] Configuring second RS232 - baud:38400 data bits:8 parity:none, stop bits:1
[17 Mar 14:36:50] Configuring RS232 - baud:38400 data bits:8 parity:none, stop bits:1
[17 Mar 14:36:50] Local time is now 14:36:50 (GMT +0)
[17 Mar 14:36:50] Starting to load project
```

Use the Controller pull-down to select the Controller whose log you wish to view. Press the Clear button to clear the log.

TPC recovery

The TPC has a built-in failsafe against firmware problems: it stores two versions of firmware. So if one copy of the firmware fails to load, or becomes corrupted due to a loss of power during a firmware reload, the other can be used instead. However, in the event that the TPC will not startup, there is a method to recover the TPC using the memory card.

Please follow these instructions carefully:

1. Remove the memory card from the TPC and insert it into your computer
2. Wipe all files on the memory card (ensure you have made any necessary backups)
3. Locate the *firmware* directory in the Designer installation directory or app bundle*
4. Copy the file *tpc.fw* from the *firmware* directory to the memory card
5. On the memory card, rename the *tpc.fw* file to *tpc.app*
6. Reinsert the memory card into the TPC and restart the TPC. The TPC will boot, but will take longer to boot than normal. Please be patient and wait for the Active LED to illuminate continuously
7. Connect to the TPC using Designer and reload the firmware as normal
8. Remove the memory card from the TPC and insert it into your computer again
9. Delete the *tpc.app* file from the memory card
10. Reinsert the memory card into the TPC and restart the TPC.

* To locate the firmware directory in the app bundle on Mac OS X, please follow these steps:

1. Navigate to your "Applications" folder located on the the Hard Drive, typically named "Macintosh HD"
2. Locate the application called "Unison Mosaic Designer"
3. Right-click (or control-click) on it and choose "Show Package Contents" from the menu that appears
4. Now navigate to Contents/Resources/firmware to find the file *tpc.fw*.

MSC Recovery

Revised MSC 1/2/4 Hardware (serial numbers 006xxx onwards)

The MSC has a built-in failsafe against firmware problems: it stores two versions of firmware. So if one copy of the firmware fails to load, or becomes corrupted due to a loss of power during a firmware reload, the other can be used instead. However, in the event that the MSC will not startup, there is a method to recover the MSC using the memory card.

Please follow these instructions carefully:

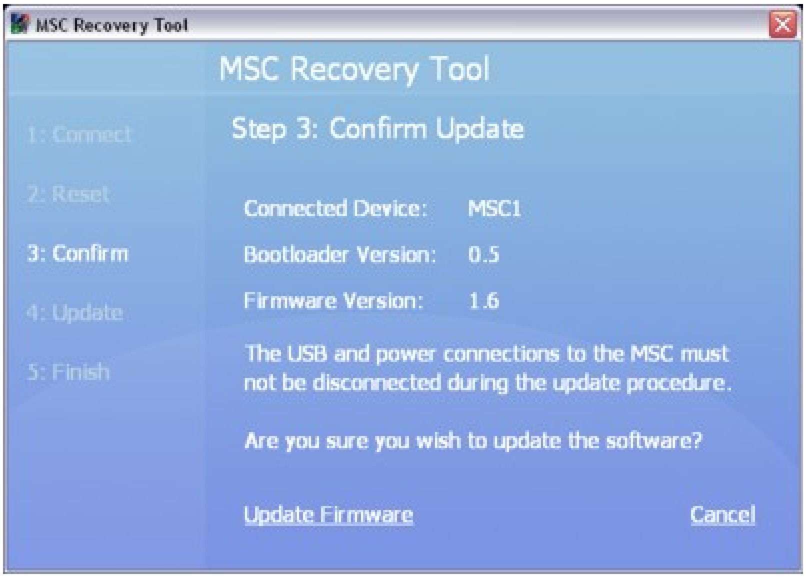
1. Remove the memory card from the MSC and insert it into your computer
2. Wipe all files on the memory card (ensure you have made any necessary backups)
3. Locate the *firmware* directory in the Designer installation directory or app bundle*
4. Copy the file *lpcr2.fw* from the *firmware* directory to the memory card
5. On the memory card, rename the *lpcr2.fw* file to *lpc.app*
6. Reinsert the memory card into the MSC and restart the MSC. The MSC will boot, but will take longer to boot than normal. Please be patient and wait for the Active LED to illuminate continuously
7. Connect to the MSC using Designer and reload the firmware as normal
8. Remove the memory card from the MSC and insert it into your computer again
9. Delete the *lpc.app* file from the memory card
10. Reinsert the memory card into the MSC and restart the MSC.

* To locate the firmware directory in the app bundle on Mac OS X, please follow these steps:

1. Navigate to your "Applications" folder located on the the Hard Drive, typically named "Macintosh HD"
2. Locate the application called "Unison Mosaic Designer"
3. Right-click (or control-click) on it and choose "Show Package Contents" from the menu that appears
4. Now navigate to Contents/Resources/firmware to find the file *lpcr2.fw*.

Original MSC 1/2 Hardware (serial numbers 001xxx - 005xxx)

The Legacy MSC Recovery Tool is a standalone utility for updating the original MSC's bootloader and firmware:



Step 3: Confirm Update

Connected Device: MSC1

Bootloader Version: 0.5

Firmware Version: 1.6

The USB and power connections to the MSC must not be disconnected during the update procedure.

Are you sure you wish to update the software?

When to use it

To update the bootloader

The Legacy MSC Recovery Tool is the only method by which the bootloader can be updated. The latest version is 0.5 and you should upgrade to this version.

To recover corrupt firmware

It is possible, although rare, for the firmware to become corrupt when updating using the Designer software, particularly over an Ethernet connection. An MSC with corrupt firmware (as indicated by the status LEDs, see troubleshooting) can not be updated using the Designer software and so the Legacy MSC Recovery Tool must be used instead.

How to use it

Connect the MSC via USB

Use either a DC supply or PoE to power the MSC and then connect to the PC with a standard USB cable. When first connecting an MSC to a Windows PC you will be prompted to install the Unison Mosaic USB drivers via the "Found New Hardware" wizard. Follow the default instructions (answering "No" then "Auto" to the prompts) to install both drivers - the process will happen twice, once for the MSC's data port and then again for the control port.

Run the tool

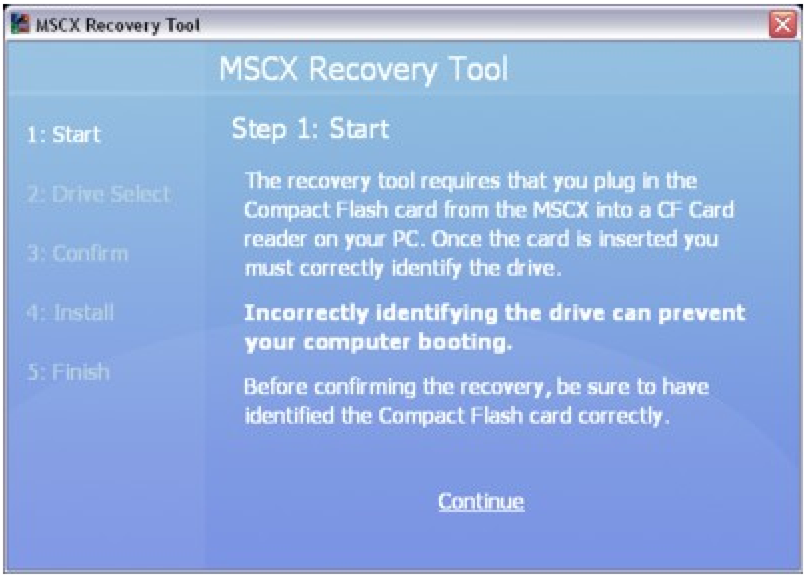
On Windows this tool can be launched from the Start menu under Programs > Mosaic > Designer. On Mac OS X the tool can be launched from wherever the Designer software was installed, typically in Applications or Desktop. Unison Mosaic Designer must be closed before launching this tool.

To update the bootloader to the latest version (0.5) press "Update Bootloader" and the process will begin, taking a few seconds to complete during which time you must not disturb the USB connection, the PC nor the MSC in any way. Interrupting this process can corrupt the bootloader image and render the MSC inoperable. Press Exit to finish and then launch the tool again to update the firmware.

To update the firmware to the latest version (v1.11) press "Update Firmware" and the process will begin, taking a few seconds to complete during which time you must not disturb the USB connection, the PC nor the MSC in any way. Should this process be interrupted or fail for some reason then try again. Press Exit to finish.

MSC X Recovery Tool

The MSC X Recovery Tool is a standalone utility for updating the MSC X's firmware:



MSCX Recovery Tool

Step 1: Start

The recovery tool requires that you plug in the Compact Flash card from the MSCX into a CF Card reader on your PC. Once the card is inserted you must correctly identify the drive.

Incorrectly identifying the drive can prevent your computer booting.

Before confirming the recovery, be sure to have identified the Compact Flash card correctly.

When to use it

To recover corrupt firmware

It is possible, although rare, for the firmware to become corrupt when updating using the Designer software. An MSC X with corrupt firmware (as indicated by the status LEDs, see troubleshooting) can not be updated using the Designer software and so the MSC X Recovery Tool must be used instead.

To install a new or larger capacity card

Should you wish to use a new or larger capacity Compact Flash card then you must first run this tool to format the card and install the MSC X firmware.

How to use it

Remove the MSC X's Compact Flash card

Disconnect the MSC X from the mains supply.

Remove the right hand front panel of the MSC X (socket screw top and bottom) and press the card release button to access the Compact Flash card.

Plug the Compact Flash card into a CF card reader on a PC

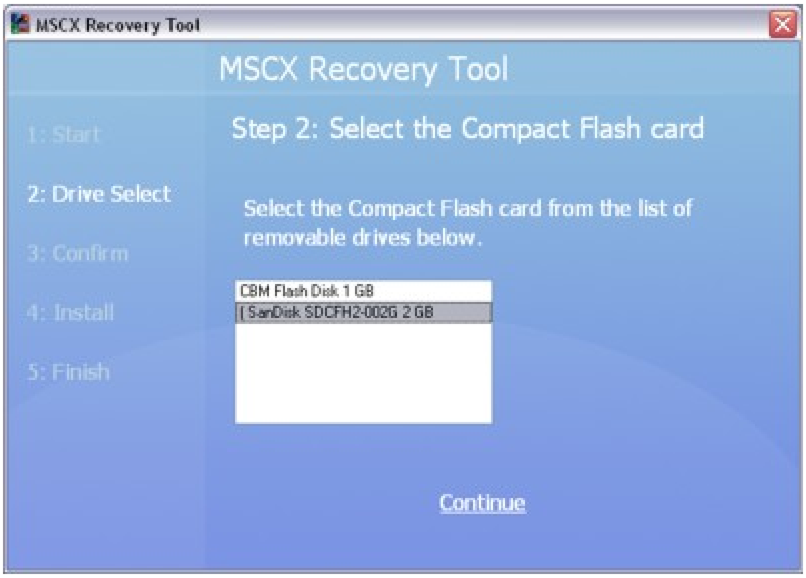
Insert the card into a suitable card reader on your PC.

Run the tool

On Windows this tool can be launched from the Start menu under Programs > Mosaic > Designer. Note that the tool has to be run as administrator and the Start Menu shortcut is thus configured accordingly. However, if you wish to run the application directly, you will need to right click on it and choose run as administrator.

On Mac OS X the tool can be launched from wherever the Designer software was installed, typically in Applications or Desktop.

Press "Continue" to select the correct Compact Flash card:



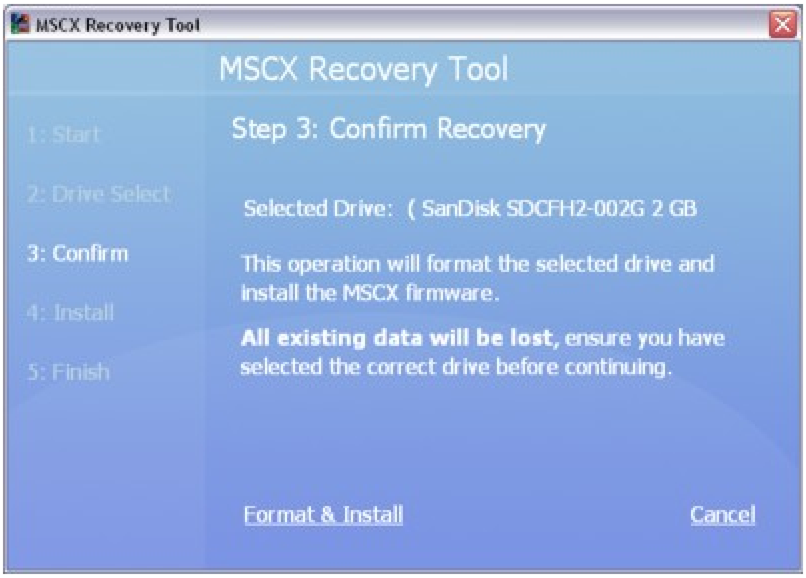
MSCX Recovery Tool

Step 2: Select the Compact Flash card

Select the Compact Flash card from the list of removable drives below.

CBM Flash Disk 1 GB
(SanDisk SDCFH2-002G 2 GB

Select the card and press "Continue" to confirm the correct Compact Flash card:



MSCX Recovery Tool

Step 3: Confirm Recovery

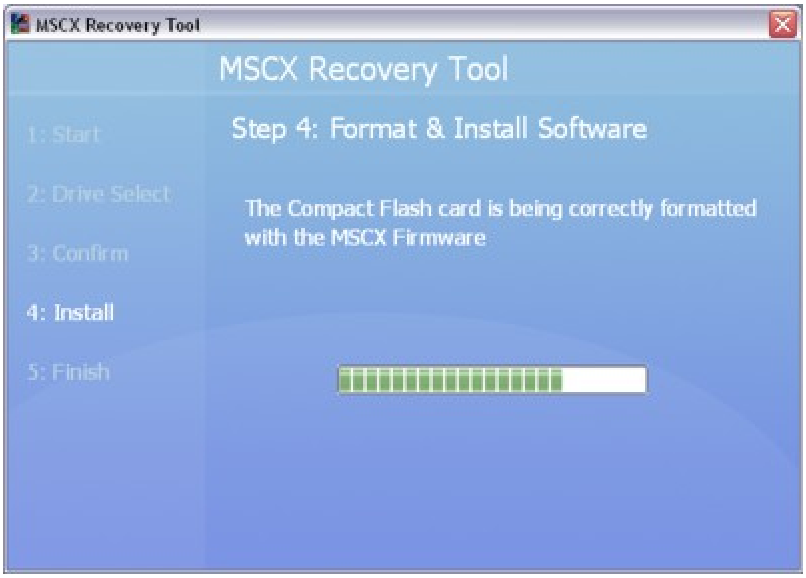
Selected Drive: (SanDisk SDCFH2-002G 2 GB

This operation will format the selected drive and install the MSCX firmware.

All existing data will be lost, ensure you have selected the correct drive before continuing.

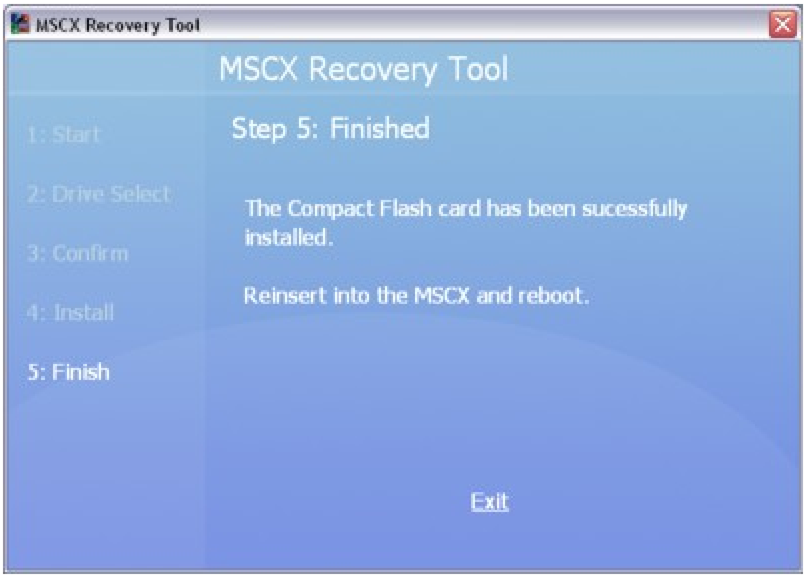
IMPORTANT: All existing data will be lost, ensure that you have selected the correct card before proceeding.

Press "Format & Install" once you have confirmed that the correct card has been selected to proceed and complete:



Step 4: Format & Install Software

The Compact Flash card is being correctly formatted with the MSCX Firmware



Step 5: Finished

The Compact Flash card has been sucessfully installed.

Reinsert into the MSCX and reboot.

Press "Exit" and eject the card from the PC (Windows; right-click and select Eject or use Safely Remove Hardware from the task bar, Mac OS X; drag to trash).

Remove the card from the reader.

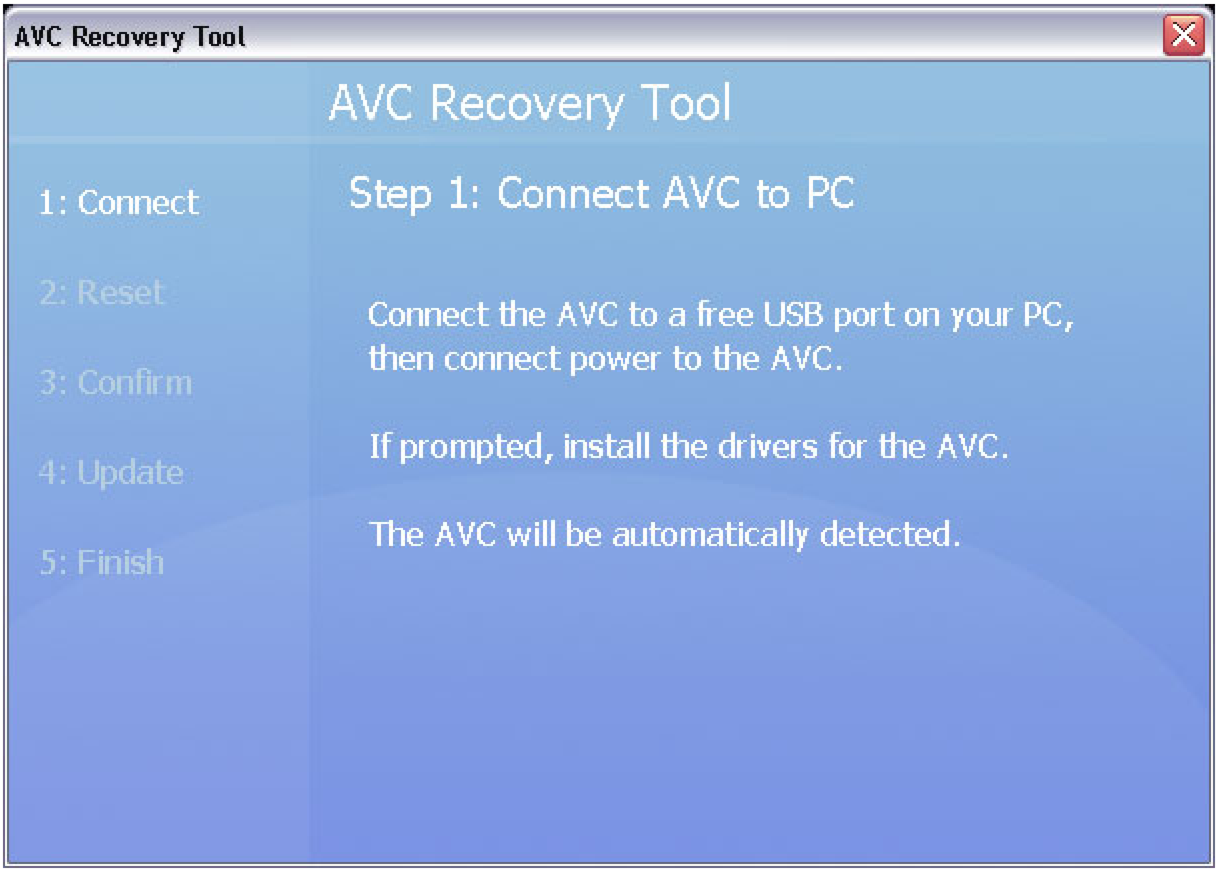
Reinsert the Compact Flash card into the MSC X & reboot

Reinsert the card into the MSC X and replace the front panel.

Reconnect the mains supply to reboot and then upload your project programming.

AVC Recovery Tool

The AVC Recovery Tool is a standalone utility for updating the firmware for both processors in the AVC. Currently this tool can only be run on Windows.



Step 1: Connect AVC to PC

Connect the AVC to a free USB port on your PC, then connect power to the AVC.

If prompted, install the drivers for the AVC.

The AVC will be automatically detected.

When to use it

To update the firmware for both processors in the AVC

The AVC has two processors. Reloading firmware from Designer updates the firmware for one of the processors (the DSP) but not the other (the MCU). Occasionally we release updates for the MCU as well as for the DSP and at this point the AVC Recovery Tool must be used.

To recover corrupt firmware

It is possible, although rare, for the firmware to become corrupt when updating using the Designer software, particularly over an Ethernet connection. An AVC with corrupt firmware (as indicated by the status LEDs, see troubleshooting) can not be updated using the Designer software and so the AVC Recovery Tool must be used instead.

How to use it

Connect the AVC via USB

Use a DC supply to power the AVC and then connect to a PC with a standard USB cable. If prompted to install the USB driver, follow the steps in the Found New Hardware wizard, selecting to install the driver automatically.

Run the tool

On Windows this tool can be launched from the Start menu under Programs > Mosaic > Designer. Unison Mosaic Designer must be closed before launching this tool.

To update the firmware to the latest version (v1.11) press "Update Firmware" and the process will begin, taking a few seconds to complete during which time you must not disturb the USB connection, the PC nor the AVC in any way. Should this process be interrupted or fail for some reason then try again. Press Exit to finish.

Software release notes

Release Notes

These are provided with the installation Readme file.

Software Licences

GPL

Portions of this software are licensed under the GNU General Public License version 2. The license is available in the file "gpl-v2-license.txt" contained in the Unison Mosaic Designer install directory.

To obtain this software either visit www.carallon.com or send a stamped self-addressed envelope containing a blank CD to:

GPL Compliance,
Carallon Limited,
Studio G20 Shepherds Building,
Rockley Road,
London W14 0DA
England

System limits & capacities

Unison Mosaic Designer imposes the following project limits which can not be exceeded:

Groups	1024	
Fixtures	30000	Discrete or compound fixtures
Fixture elements	60000	Elements or pixels within compound fixtures eg. 18 per James Thomas Pixeline 1044
Moving lights	200	Automated or moving lights and conventional fixtures with scrollers
Pixel matrices	256	
Frame arrays	1000	Instances of media, perlin noise, starfield, text and custom presets deployed on timelines
Timelines	500	
AVC presets	256	
Custom presets	256	
Media presets	256	Imported media clips
Media slots	1024	
Text slots	1024	
Fonts	16	
Mover presets	256	
Triggers	1024	
Conditions per trigger	32	
Actions per trigger	32	
Controllers	40	So if all MSC 2s then the maximum number of DMX universes in the project is 80 (80 x 512 = 49960 DMX 512 channels)
Remote Devices (excluding RIO D)	100	Per Remote Device type per Designer project. If you are planning to use more that 16 of one or more remote devices please contact support to discuss your requirements in advance.
DALI and RIO Ds		Please see DALI Interfaces for DALI and RIO D limitations
KiNet power supplies	1024	Per Controller in a Designer project
Timecode Buses	6	MIDI or linear (SMPTE/EBU) timecode sources
Network Buses	5	Ethernet trigger sources eg. UDP
Audio Buses	4	Audio trigger sources
Plan size (pixels)	8192x8192	Plan and fixture library scale is 1cm:1pixel (0.394":1pixel)

As you can see from the above limits, the Unison Mosaic control system can scale to an impressive size that rivals even state-of-the-art lighting consoles.

However, while the Controllers themselves provide a scalable solution (each Controller only stores and processes the data relevant to it), the PC running the Designer software will need greater performance, particularly the OpenGL graphics system, as the project size increases. For very large projects, or projects where some of the above limitations are restrictive, please contact support to discuss your requirements in advance.

Best Practices

Just like any other computational device, Controllers have a finite amount of resources available to them.

Triggering and playback expectations

The Unison Mosaic system is designed to spread the load across all Controllers in a project. You can aid this by patching fixtures as evenly as possible across all Controllers.

As well as lighting playback, Controllers are often used as interfaces to a wider system. This was intentional and is why the system is so flexible but bear in mind that if a Controller is being asked to deal with a substantial amount of incoming triggers or outgoing actions (such as Ethernet or serial communications) then this may have a negative impact on show playback and system responsiveness. If you have any questions about this, please contact support with your project requirements and we'll be happy to advise.

Looping and holding at end

If you need a timeline to run continuously, we'd always recommend setting it to hold at end (rather than loop) where appropriate. A looping timeline requires significantly more processing power because the Controller has to track the time position until released. You should always release timelines when they are not actively affecting output. More information about the differences between loop and hold at end can be found here.

Transparency

Transparency is a very processor-intensive effect for Controllers to generate. Having one or two layers of transparency on some fixtures will be fine, but having several layers on a fully patched MSC 4 may cause some playback to be choppier than normal.

AVC Timeline Restrictions

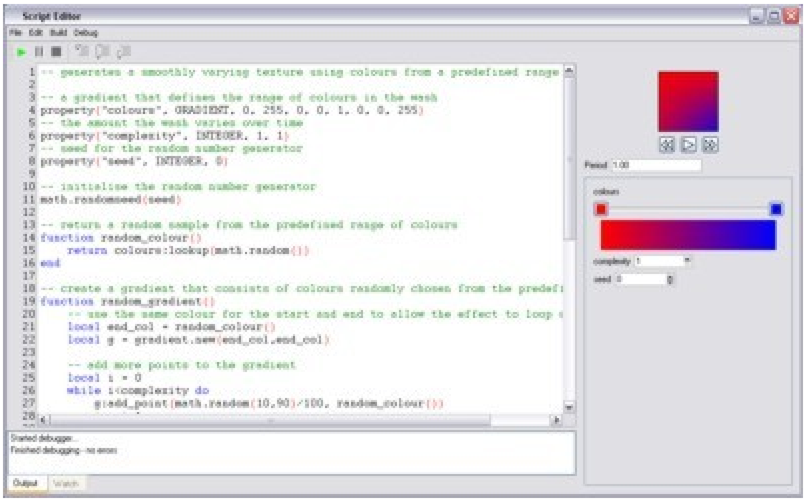
AVCs don't support timelime pausing, position setting or playback rate changes.

Lua scripts

We use a scripting language called Lua, which has been extended to provide functionality specific to the Unison Mosaic Controllers. Tutorials and reference manuals for the Lua language can be found at www.lua.org. We will not attempt to document the Lua language here, but just the Unison Mosaic specific extensions. Please contact support if you need assistance with preparing a script or if you would like some examples as a starting point.

Lua script editor

The Lua Script Editor allows you to edit scripts from Triggers Conditions and Actions, and Custom presets within Designer. The Script Editor is launched by pressing the Launch Editor button in the trigger properties:



```
-- generates a smoothly varying texture using colours from a predefined range

-- a gradient that defines the range of colours in the wash
property("colours", GRADIENT, 0, 255, 0, 0, 1, 0, 0, 255)
-- the amount the wash varies over time
property("complexity", INTEGER, 1, 1)
-- seed for the random number generator
property("seed", INTEGER, 0)

-- initialise the random number generator
math.randomseed(seed)

-- return a random sample from the predefined range of colours
function random_colour()
    return colours:lookup(math.random())
end

-- create a gradient that consists of colours randomly chosen from the predef
function random_gradient()
    -- use the same colour for the start and end to allow the effect to loop
    local end_col = random_colour()
    local g = gradient.new(end_col,end_col)

    -- add more points to the gradient
    local i = 0
    while i<complexity do
        g:add_point(math.random(10,90)/100, random_colour())
```

```
Started debugger...
Finished debugging - no errors
```

The main area of the editor is the code editor where you enter the source code of the script. The code editor will colour the Lua syntax to aid readability. Standard clipboards shortcuts and undo/redo are supported.

To import a Lua script from an external file, use File > Import (Ctrl+I).

To save a Lua script to a file, use File > Export (Ctrl+E).

To compile the script and check for syntax errors, use Build > Build (F7). If there are errors in the script, they will be displayed at the bottom of the window.

When you have finished editing the script, press Ok to close the Script Editor. If there are still errors in your script, you will be prompted whether the invalid source should be saved or not.

To discard any changes close the Script Editor and select Discard from the confirmation dialog.

Debugging Lua scripts

If your script contains no errors, use Debug > Run (F5) to execute it in a debugger. This will run the script and report any runtime errors in the Output tab below the source editor. Note that only Custom preset scripts can be executed in the debugger.

If there are errors, you can place breakpoints in the source to force the debugger to halt execution at that point and allow you to examine the state of variables and execute further instructions one at a time.

To add/remove a breakpoint on a line of source code, click the mouse to the left of the relevant line number, or use Debug > Toggle Breakpoint (F9) to set a breakpoint at the text cursor. Now, when you run the script with Debug > Run (F5), the debugger will halt at the first breakpoint that it encounters.

When the debugger is halted, you can examine the values of variable in the script in the Watch tab below the source editor.

- To execute the next line of code, use Debug > Step Over (F10).
- If the debugger is halted at a function call, use Debug > Step Into (F11) to step into the function.
- To step out of the current function, use Debug > Step Out (Shift+F11).
- To resume normal execution, use Debug > Run (F5).
- To terminate execution immediately and return to editing the source, use Debug > Stop (Ctrl+F5).
- While a script is executing, use Debug > Break to halt execution at the current line.

Unison Mosaic language extensions for triggers

Accessing the realtime clock (local to this Controller)

realtime.year	The year
realtime.month	The month (1 to 12)
realtime.monthday	The day of the month (1 to 31)
realtime.weekday	The day of the week (0 = Sunday; 1 = Monday; etc.)
realtime.hour	The hour (0-23)
realtime.minute	The minute (0-59)
realtime.second	The second (0-59)

NOTE: The location of the Controller must be set correctly in project properties for these to return the correct values.

Accessing the calculated sunrise and sunset times (local to this Controller)

sunrise.hour	The hour of sunrise (0-23)
sunrise.minute	The minute of sunrise (0-59)
sunrise.second	The second of sunrise (0-59)
sunset.hour	The hour of sunset (0-23)
sunset.minute	The minute of sunset (0-59)
sunset.second	The second of sunset (0-59)

NOTE: The location of the Controller must be set correctly in project properties for these to return the correct values.

Accessing the calculated twilight times (local to this Controller)

civil_dawn.hour	The hour of civil dawn (0-23)
civil_dawn.minute	The minute of civil dawn (0-59)
civil_dawn.second	The second of civil dawn (0-59)
civil_dusk.hour	The hour of civil dusk (0-23)

civil_dusk.minute	The minute of civil dusk (0-59)
civil_dusk.second	The second of civil dusk (0-59)
nautical_dawn.hour	The hour of nautical dawn (0-23)
nautical_dawn.minute	The minute of nautical dawn (0-59)
nautical_dawn.second	The second of nautical dawn (0-59)
nautical_dusk.hour	The hour of nautical dusk (0-23)
nautical_dusk.minute	The minute of nautical dusk (0-59)
nautical_dusk.second	The second of nautical dusk (0-59)

NOTE: The location of the Controller must be set correctly in project properties for these to return the correct values.

TPC user interface control

These are only executed on the local Controller, which must be a TPC.

set_control_value(controlKey, valueIndex, value)	Sets the value of the TPC control. Control key is a string, e.g. "slider002". Value index 1 for sliders or 1, 2 or 3 for colour pickers
set_control_value(controlKey, value)	Shorthand form with value index equal to 1, useful for sliders. Control key is a string, e.g. "slider002"
set_control_state(controlKey, stateName)	Provide feedback by changing the appearance of a control. Control key is a string, e.g. "slider002".
set_page(pageName)	Change the current page of the user interface
set_locked(lock)	Lock or unlock the TPC (lock: true of false)
set_enabled(enable)	Enable or disable the user interface (enable: true of false)

Accessing the current state of the digital inputs on this Controller

digital[index]	Boolean value on input (index 1 to 8)

Accessing the values of DMX inputs on the local Controller

DMXIN[channel]	The value of the specified channel (1 to 512)
get_dmxin(x)	Returns a DMXIN object for port (x is 1 or 2)

Checking the local Controller number

get_controller_number()	Returns the number of this Controller

Checking the status of timelines (local to this Controller)

is_timeline_running(num)	Is timeline with specified user number between start and end time?
is_timeline_onstage(num)	Is timeline with specified user number actually in control of fixtures on this Controller?

Injecting a trigger

Equivalent to clicking a trigger in the web control screen.

inject_trigger(num)	Fires the trigger with the specified number. Additional arguments are passed as variables.

Timeline control

Note that unlike a normal trigger action these will only affect the local Controller. This can be a powerful additional feature in certain situations. If you want to fire a timeline system-wide then inject an appropriate trigger.

start_timeline(num)	Start timeline with the specified user number
stop_timeline(num, time)	Stop timeline with the specified user number with optional fade time (in seconds)
halt_timeline(num)	Halt timeline with the specified user number
resume_timeline(num)	Resume timeline with the specified user number
stop_all()	Stop all timelines

Timecode control (local to this Controller)

set_timecode_source_enabled(source, [enabled])	Enable or disable a timecode source (source is one of `TCODE_1`, `TCODE_2`, `TCODE_3`, `TCODE_4`, `TCODE_5`, `TCODE_6`, `AUDIO_1`, `AUDIO_2`, `AUDIO_3`, `AUDIO_4`; enabled assumed true if not specified)
set_timecode_source(timeline_num, source, [offset])	Set the current timecode source for a timeline (source is one of `DEFAULT`, `TCODE_1`, `TCODE_2`, `TCODE_3`, `TCODE_4`, `TCODE_5`, `TCODE_6`;offset in msecs, assumed 0 if not specified)
set_timecode_source(timeline_num, source, band, channel, [peak])	Set the current audio source for a timeline (source is one of `AUDIO_1`, `AUDIO_2`, `AUDIO_3`, `AUDIO_4`; band can be any number or `VOLUME`; channel is one of `LEFT`, `RIGHT`, `COMBINED`; peak is optional, assumed false if not specified)

Fixture control (local to this Controller)

A script can directly control fixtures as if it was a high priority timeline. It can only control fixtures local to the Controller on which the script is running. It can also only control whole fixtures – there is no access to individual elements of compound fixtures.

set_intensity(fixture, value, time)	Set intensity on fixture to value (as DMX). Optionally give a time (in seconds) to fade.
set_red(fixture, value, time)	Set red on fixture to value (as DMX). Optionally give a time (in seconds) to fade.
set_green(fixture, value, time)	Set green on fixture to value (as DMX). Optionally give a time (in seconds) to fade.
set_blue(fixture, value, time)	Set blue on fixture to value (as DMX). Optionally give a time (in seconds)

	to fade.
clear_fixture(fixture, time)	Clear the direct settings on fixture with optional fade time (in seconds).
clear_all(time)	Clear all direct settings with optional fade time.

Accessing the values of DMX channels (local to this Controller)

DMXOUT[channel]	The value of the specified channel (1 to 512)
get_dmxout(universe)	Returns a DMXOUT object for universe (see table below)

Park and Unpark DMX channels (local to this Controller)

park(universe, channel, value)	Park the specified channel (1-512) on universe (see table below) at value (0-255).
unpark(universe, channel)	Unpark the specified channel (1-512) on universe (see table below).

The value for the universe argument for both of the above functions depends on the protocol type:

MSC 1's DMX	Universe = 1.
MSC 2's DMX	Universe = 1 or 2.
Controller's Art-Net II	Universe = ARTNET+<Art-Net II universe number>.
Controller's Pathport	Universe = PATHPORT+<Pathport universe number>.
Controller's ETCNet2	Universe = NET2+<ETCNet2 universe number>.
Controller's sACN	Universe = SACN+<sACN universe number>.
Controller's KiNet	Universe is obtained using the get_kinet_universe(powerSupplyNum, portNum) function.

For example:

park(ARTNET+2, 1, 255)	Parks channel 1 of Art-Net II universe 2 at full.
park(ARTNET, 2, 128)	Parks channel 2 of Art-Net II universe 0 at 50%.
park(SACN+1, 1, 255)	Parks channel 1 of sACN universe 1 at full.
park(get_kinet_universe(2, 15), 1, 255)	Parks channel 1 of port 15 of KiNet power supply 2 at full.

Accessing the current state of the digital inputs on a RIO

get_rio(type, number)	Returns as an object the RIO by type (RIO_80, RIO_44 or RIO_08) and number.
rio[input]	Boolean value on input (1 to 8) of the returned RIO object.

For example:

```
myRIO = get_rio(RIO_80, 2)
rioInput1 = myRIO[1]
```

Accessing the current state of BPS buttons and setting BPS LEDs

get_bps(number)	Returns as an object the BPS by number.
bps:get_state(button)	State of the button (1 to 8) of the returned BPS object, values are RELEASED, PRESSED, HELD or REPEAT.
bps:set_LED(button, effect, intensity, fade)	Sets the button (1 to 8) of the returned BPS object, effect can be OFF, STATIC, SLOW_FLASH, FAST_FLASH, DOUBLE_FLASH, BLINK, PULSE, SINGLE, RAMP_ON or RAMP_OFF. Intensity is optional and should be between 0 and 255, if not specified, the LED intensity is left unchanged from its current value. If intensity is specified, fade can also optionally be specified and should be between 0s and 25.4s.

For example:

```
myBPS = get_bps(1)
if (myBPS:get_state(2) == HELD) then
    mpBPS:set_LED(2, FAST_FLASH, 255, 0.5)
end
```

Accessing trigger variables

The variable must have been captured by the trigger running the script.

variable[index]	Value of variable with given index

Writing messages to the Controller's status log

log(“string”)	Writes string into the log

Storing data on the memory card

Using a Lua script it is possible to store data as a file on the memory card. Unlike variables in memory, data stored to a file will not be affected by resetting the Controller or cycling the power. The file can be read back at any time to access the data. This can be very useful in show control situations where some system status information needs to be stored persistently even if the Controller is reset. Alternatively it can allow you to store large data tables on the memory card and only read the data when it is needed.

To work with files on the memory card use the standard I/O library functions described in the Lua documentation. When writing to a file always close the file explicitly to make sure your data is correctly stored. The full file path to a file on the memory card differs per Controller, so we have provided a function to create the full file path for you:

get_resource_path("my_file")	Returns the full file path for the given file name.

For example:

```
file = io.open(get_resource_path("myfile"), "r")
if (file ~= nil) then
    local line = file:read()
    file:close()
end
```

Trigger Programming Guide

Introduction

The Unison Mosaic Controllers offer many useful show control capabilities. Frequently it is the ability to cope with the particular show control needs of a project that is the critical factor in selecting a control system.

Show control broadly consists of two tasks. First we need to be able to interface with other devices, which may either be triggering us or be under our control. The Unison Mosaic Controller supports most of the core interfaces typically used for show control, either directly on the unit (contact closures, RS232, MIDI, TCP/IP, time and date) or via Expansion Modules (RS485, SMPTE) or Remote Devices. Within the Triggers screen of the Designer software we can configure the Controller to detect particular triggers and how to respond to them.

Second we need to be able to make decisions. These could be simple choices between two alternatives - perhaps a contact closure needs to trigger a different timeline depending on whether it is during the day or during the night. Within the Triggers screen we support a range of conditions that can be used to quickly implement this sort of logical decision making. We also provide a facility to treat values received on an input as a variable that can be used to alter the behaviour of actions - such as using a number received via RS232 to select a particular timeline.

The standard capabilities offered in the Triggers screen are extensive, but a good show control system has the ability to cope with situations that are anything but standard. Within the Unison Mosaic system when things get non-standard then we can use scripting.

Script is a simple programming language that allows users to extend the functionality of the Unison Mosaic system themselves. We use a freely available programming language called Lua. Anyone who has ever worked with a programming language will find all the typical tools are available, and it should be straightforward to pick up for those who have not. On top of the core Lua syntax we have added some dedicated Unison Mosaic functions that allow scripts to work directly with the capabilities of a Controller.

Not every problem requires script, but there are few show control problems that can't be solved using script where necessary. A few examples of situations where you might want to use script include:

- Making a single contact closure start a different timeline each time
- Make a timeline loop a set number of times and then release
- Track motion sensor activity over a period of time
- Inverting a DMX input before it is used with a Set Intensity action
- Interpreting data from a wind direction sensor
- Using a table of times for high and low tide to control bridge lighting
- Implementing an interactive game for a science museum

We will use some of the situations as examples below.

The Basics

There are a few basic things you need to know straight away. If any of them are not immediately clear then don't worry - there are lots of examples of how to apply them in the following section.

Lua scripts are written as simple text files using any text editor. It is standard practice to use a .lua filename extension though this is not required. These text files can be loaded directly into the script editor dialog within Designer.

Comments

It is good practice to include readable comments in your scripts so that you (or anyone else) will be able to easily tell what you were aiming to achieve. In Lua everything after two dashes on a line is treated as a comment.

```
-- This is a comment
This = is + not - a * comment -- but this is!
```

The whole point of comments is that they have no effect on the behaviour of the script. But I am introducing them first so that I can use them within the examples that follow.

Variables

If you want to store a piece of data - whether it is a number, some text or just true or false - then you use a variable. You create a variable simply by giving it a name and using it in your script. A variable can store any type of data just by assigning it.

```
firstVariable = 10 -- assign a number
anotherVariable = "Some text" -- assign a string
```

When you next use these names then they will have the values that you assigned to them:

```
nextVariable = firstVariable + 5 -- value of nextVariable will be 15
```

Note that names are case-sensitive (i.e. capitals matter!), and once you have named a variable once then any time you use the same name you will be referring to the same variable - in programming terms it is *global*. This even applies across different scripts - so you can assign a number to a variable called `bob` in one script and then use the number in another script by referencing `bob`.

One of the most common errors when writing scripts is trying to use a named variable before it has been assigned a value - this will result in an error when the script is run. It is also very easy to use the same name in two different places and not realise that you are actually reusing a single variable. (There is a way of dealing with this for names you want to reuse that we will touch on later.)

Arithmetic

Scripts will often need to do some arithmetic - even if it is something very basic like keeping a counter of how many times it is run:

```
myCount = myCount + 1
```

All of the standard arithmetic operations are available. There is also a library of mathematical functions available should it be required, which includes things like random number generators.

Flow of Control

In most scripts there will be one or more points where you want to make choices. Lua provides four useful structures for this. The most common is `if`, where you can choose which path to take through the script by performing tests.

```
if myNumber < 5 then
```

```
    -- first choice
elseif myNumber < 15 and myNumber > 10 then
    -- second choice
else
    -- third choice
end
```

The other control structures all involve blocks of script that need to be repeated a certain number of times. The most straightforward is the `while` loop, which will repeat the enclosed block of script as long as the test at the start is true:

```
myNumber = 10
while myNumber > 0 do
    -- some useful script
    myNumber = myNumber - 1 -- myNumber counts down
end
```

The `repeat until` loop is really exactly the same, but here the test is done at the end of each loop and it will repeat while the test is false.

```
myNumber = 1
maxNumber = 4096
repeat
    -- some useful script
    myNumber = myNumber * 2
until myNumber == maxNumber
```

Here it is worth noting the use of two equal signs == to mean 'is equal to' in a test. This is different from a single equal sign, which is used for assigning values. It is another very common mistake to assign a value when you meant to test if it was equal, and it can be hard to spot because it is valid syntax that will not generate an error. The opposite of == meaning 'is equal to' is ~= meaning 'is not equal to'.

The other control structure is the `for` loop, which has a number of powerful options beyond the scope of what we need here. But it is worth seeing how it can be used to do basic loops in a slightly neater way:

```
for i = 1,10 do
    -- some useful script where i has value 1 to 10
end
```

A final word of caution regarding loops: be careful that you do not write a loop that will never exit! This is all too easy to do by forgetting to increment a counter value that you are using in the test for the loop. If your script has one of these 'infinite loops' then the Controller will get stuck when it runs the script and be reset by the watchdog feature (provided this is enabled). Script is a tool for the grown-ups and it will not protect you from doing silly things - so make sure you test your scripts carefully before leaving them to run.

Tables

Often you will need to store a set of values within a script - these might be a list of timeline numbers or the current states of all the contact closure inputs. Lua allows us to store multiple values within a single named variable and this is called a Table.

A table has to be created before it can be used:

```
firstTable = {} -- creates an empty table
secondTable = { 5,3,9,7 } -- a table with 4 entries
```

You can then access entries within the table by indexing into it - signified by square brackets. The number within the square brackets identified which entry within the table you want to use or modify.

```
x = secondTable[3] -- x now equals 9 (3rd entry)
firstTable[1] = 5 -- entry 1 now has value 5
firstTable[7] = 3 -- entry 7 now has value 3
x = firstTable[1] + firstTable[7] -- x now equals 5 + 3
```

Note that we are allowed to assign values to entries within the table without doing anything special to change the size of the table. We can keep adding elements to the table as needed and Lua will take care of it for us. This makes it possible to write scripts using tables that will work regardless of how many entries there are in the table (e.g. a list of 4 timeline numbers or of 40).

Tables are particularly powerful when used together with the loops we looked at in the previous section. For example if I have a table of numbers and I wanted to find the smallest then I could use the following script:

```
numbers = { 71,93,22,45,16,33,84 }

smallest = 10000 -- initialise with large number
i = 1 -- use to count loops
while numbers[i] do
    if numbers[i] < smallest then
        smallest = numbers[i]
    end
    i = i+1
end
```

This is our first really functional piece of script and there are a couple of things worth noting.

- The first entry in a table is accessed using the number one (i.e. `myTable[1]`). This may seem obvious - but some other programming languages start counting from zero.
- As we increment the variable `i` each time around the loop this means we will be looking at a different entry in the table each time around. The test at the start of my while loop is written to work regardless of how many entries there are in the table. When you use a table entry in a test like this then it will be true as long as the entry has some value (even if the value is zero) and false if there is no value there at all.

Functions

Within script there are a whole range of pre-defined operations that you can call when writing your own scripts. Some of these are provided by the Lua language and are fully described in its documentation. Others have been provided by ETC to allow you to interact with the Controller from script and are fully described in the manual. They are all called functions and accessed using a similar syntax. For example:

```
x = math.random(1,100)
```

This will assign variable `x` a value that is a random number between 1 and 100. The function `math.random()` is a standard function provided by Lua and we can control its behaviour by passing in an argument - in this case the values 1 and 100 to tell it the range within which we want our random number to fall.

```
t = 5
```

```
start_timeline(t)
```

`start_timeline` is one of the functions provided by ETC and it will start the timeline with the number passed in as an argument.

It is also possible to define your own functions as part of script. You might do this if there is a block of script that you know you will need to reuse in a lot of different places. It will be much easier to write the script in one place and then call it from wherever you need it.

```
function diff(a, b)
    if a > b then
        return a - b
    else
        return b - a
    end
end

v1 = 10
v2 = 6
v3 = diff(v1,v2)
```

Note that the script containing the function definition must have been run before we try to call the function. It is often useful to have a script that is run by the Controller startup trigger which defines your functions and creates any tables - other scripts that are run by triggers can make use of those functions and tables.

Practical Examples

In this section we will go through a number of practical examples of how script can be used with a Controller. These examples are all based on real projects that are installed and working. They do get progressively more involved, so do not worry if you don't follow the later ones - you will still be able to use script successfully to solve many problems.

If you are working through this document on your own then look out for where I ask a question and rather than reading straight on I recommend stopping and trying to answer it yourself. You will only get truly comfortable with writing scripts by doing it!

Cycling through different timelines

We are installing a wall of RGB LED fixtures in a children's play area. There is a single large button that the kids are supposed to press. Each time they press it they should get a different colour or effect on the wall.

Each colour or effect would be programmed as a different timeline in Designer. The button will connect to a contact closure and so we will have a single Digital Input trigger. Rather than starting a timeline directly we will instead run the following script:

```
-- which timelines should we cycle through?
timeline = { 22, 14, 24, 16, 15, 17, 21 }
n_timeline = 7

-- on first time of running, initialise index
if not index then
```

```
    index = 1
end

-- start the timeline whose number is at entry 'index'
start_timeline(timeline[index])

-- increment index
index = index + 1

-- should we go back to the beginning of the table?
if index > n_timeline then
    index = 1
end
```

How would this change if we wanted each button press to choose a timeline at random rather than cycling through them in order?

```
-- which timelines should we cycle through?
timeline = { 22, 14, 24, 16, 15, 17, 21 }
n_timeline = 7

-- use the random function to set index
index = math.random(1,n_timeline)

-- start the timeline whose number is at entry 'index'
start_timeline(timeline[index])
```

Of course if the timeline selection is truly random then it will sometimes select the same timeline twice in a row. If we wanted to prevent this from happening how could we do it?

```
-- which timelines should we cycle through?
timeline = { 22, 14, 24, 16, 15, 17, 21 }
n_timeline = 7

-- find an index different from the old one
while index == oldIndex do
    -- use the random function to set index
    index = math.random(1,n_timeline)
end

-- store the index for next time round
oldIndex = index

-- start the timeline whose number is at entry 'index'
start_timeline(timeline[index])
```

Make a timeline loop N times

The designer has requested that a particular timeline runs once at sunset on a Monday, but twice at sunset on a Tuesday, three times at sunset on Wednesday, etc. He is planning to keep changing the timeline so does not want to have lots of copies.

There are actually lots of perfectly reasonable ways to solve this using script. Let's assume we have a single astronomical clock trigger that fires at sunset and runs the following script:

```
N = realtime.weekday -- 0 is Sunday, 1 is Monday,...

-- we want Sunday to be 7 rather than 0
if N == 0 then
    N = 7
end

start_timeline(1)
```

The timeline would be set to loop when it was programmed. We also put a flag on the timeline at the end and make a flag trigger that runs a second script:

```
-- decrement N
N = N - 1

if N == 0 then
    -- release timeline 1 in time 5s
    stop_timeline(1,5)
end
```

Note how this works by setting the value of the variable `N` in one script and then using that variable in another script, which is often a useful technique.

I have used two scripts here, but it is possible to do the same job using only one - can you see how?

In this case you would have the sunset trigger start the timeline directly and use the following script on the flag trigger:

```
-- is this the first time round?
if not N or N == 0 then
    N = realtime.weekday -- 0 is Sunday, 1 is Monday,...

    -- we want Sunday to be 7 rather than 0
    if N == 0 then
        N = 7
    end
end

-- decrement N
N = N - 1

if N == 0 then
    inject_trigger(2) -- runs action on trigger 2
end
```

The trick here is to detect whether it is the first time round the loop - if the Controller has started up today then `N` will have no value and so `not N` will be `true`, otherwise `N` will have been left with the value zero when the script ran yesterday. When we detect it is the first time then we set its initial value in the same way as before.

I have also used a different method to do the timeline release. Rather than calling `stop_timeline` directly from the script I am causing trigger number 2 to fire. We can then configure trigger number 2 to have an action that releases the correct timeline. It is sometimes easier to write scripts like this when they are going to be sent out to a customer who does not know how to modify them. In this case all the customer needs to know is to modify the start and release timeline actions in the trigger window if they want to change which timeline is run - they do not need to modify the script.

Track motion sensor activity over a period of time

A foyer has 8 pressure pads under the carpet connected to the contact closure inputs of the Controller. We need to count how many times the pressure pads are activated in any 15 second period as a simple measure of activity in the foyer. One of 4 timelines should be selected based on the level of activity.

Hopefully by now you have a pretty good idea of how you could keep a count of the number of digital inputs using script. The new element here is a need for a 15 second timer. We don't do this using script alone but make use of the timeline facilities the Controller already offers.

First the easy bit - for each digital input there is a trigger and they all run the same very simple script:

```
count = count + 1
```

Create a timeline that has no lighting programming but has a flag at 15 seconds. We set the timeline to loop and add a startup trigger that runs it. A trigger on that flag will now fire every 15 seconds while the Controller is running and we set it to run the following script:

```
-- make sure that count has a value (not first time)
if count then
    -- decide which of the 4 timelines to run
    if count < 5 then
        start_timeline(1)
    elseif count < 10 then
        start_timeline(2)
    elseif count < 15 then
        start_timeline(3)
    else
        start_timeline(4)
    end
end

-- now reset count
count = 0
```

Ideally we should also run this script on startup to initialise `count` - otherwise any digital inputs during the first 15 seconds will try to use `count` before it has a value and the script will fail (this will not harm anything or cause the Controller to fail - it will just leave rude messages in the activity log).

Initially we have got the timer timeline running continuously. What if we only wanted to count for 15 seconds from the first digital input and then stop and wait?

First we would modify the timer timeline (say it is number 5) so that it no longer loops. Then we would modify our script on the digital inputs to be:

```
if count == 0 then
    start_timeline(5)
end

count = count + 1
```

Inverting a DMX input before it is used with a Set Intensity action

A client's existing DMX control system is connected to the Controller's DMX Input. Channel 12 controls the houselights and they want the intensity of our LED installation to increase as the houselights go to black out and to decrease as the houselights come on to full.

We can configure a DMX Input trigger so that it fires any time the value of channel 12 changes. As a side effect the DMX Input trigger will always capture the value of the channel it is watching as a trigger variable. If the client wanted the intensity of the LEDs to vary with the houselights we could put a Set Intensity action directly onto this trigger and take the level from the input. However the client wants an inverse relationship - so instead we will put a simple script in between to invert the value.

The DMX input trigger will run the following script:

```
-- get the value of the DMX channel from variable
chan = variable[1]

-- invert it
chan = 255 - chan

-- now pass it to a second trigger
inject_trigger(2,chan)
```

Trigger 2 can be a soft trigger, but its action should be Set Intensity taking a level from the input. The `inject_trigger` function allows us to pass additional arguments (after the trigger number) that will be available to the action as trigger variables.

Interpreting data from a wind direction sensor

A Controller is controlling an LED façade on all four sides of a tower block. A wind direction sensor is connected to the Controller via RS232. Every second it sends the character 'X' then a 3 digit number to the Controller which is the wind direction in degrees. The client wants the windward side of the building to always be red, the leeward side to be blue and the other two sides to be green.

We'll create four timelines that correspond to the correct lighting for the wind on each face of the building. We then need an RS232 input trigger that matches the three digit decimal number as a wildcard so that it will be stored as a trigger variable. (The wildcard for this would be `X<3d>` - for more information on this refer to the manual.) Whenever the RS232 input trigger makes a successful match it will run the following script:

```
-- get the value in degrees
dir = variable[1]

-- which face is getting the wind?
if dir < 90 then
    start_timeline(1)
elseif dir < 180 then
    start_timeline(2)
elseif dir < 270 then
    start_timeline(3)
else
    start_timeline(4)
end
```

Of course here I have assumed that the corners of the building neatly line up with 0°,90°,180° and 270°. What if they are at 45°,135°,225° and 315°?

```
-- get the value in degrees
dir = variable[1]

-- which face is getting the wind?
if dir < 45 or dir >= 315 then
    start_timeline(1)
elseif dir < 135 then
    start_timeline(2)
elseif dir < 225 then
    start_timeline(3)
else
    start_timeline(4)
end
```

As the wind direction sensor is sending us data every second, when the wind direction is very near a corner then the lights are changing back and forth a lot. The client doesn't like this and wants it limited so that it will only change at most every minute. How should we do that?

Lots of ways to solve this, but one option using script would be:

```
-- get the current minutes
now = realtime.minute

if now == lastMinute then
    return -- this will exit the script early
end

-- store in lastMinute, then continue as before
lastMinute = now

-- get the value in degrees
dir = variable[1]

-- which face is getting the wind?
if dir < 45 or dir >= 315 then
    start_timeline(1)
elseif dir < 135 then
    start_timeline(2)
elseif dir < 225 then
    start_timeline(3)
else
    start_timeline(4)
end
```

But actually a better solution in this case would probably use a dummy timeline that runs for 1 minute (place a flag at 1 minute to force its length). Add a 'timeline is running' condition to the RS232 input trigger that only allows the trigger to fire if the dummy timeline is not running. And add a second action to the trigger that starts the dummy timeline. This will ensure that the script is only allowed to run once every minute.

This is probably the better solution because it avoids having a script running every second for no reason. There is some overhead involved in running scripts and it is best to keep the number of scripts that run to the minimum necessary - particularly if the Controller is heavily loaded doing lighting effects at the same time.

Using a table of times for high and low tide

A Controller is controlling the lighting on a bay bridge. The client wants a lighting effect to run at high tide and another effect at low tide. The client has provided tide tables for the entire year and plans to update the tables each year.

A lot of the fun in this situation is in converting the data from the format in which it is provided into a Lua table. This is usually an exercise in Excel and search and replace tools, which I won't cover here. So let's assume we have generated a Lua file of the form:

```
high_tides = {
    07,11,08,13,45, -- 13:45, 8 November 2007
    07,11,09,00,36, -- 00:36, 9 November 2007
    07,11,09,11,21, -- 11:21, 9 November 2007
    etc...
}
```

It's best to put large tables like this in their own separate files and run them as separate scripts on startup. Loading a large table like this into memory will take a noticeable amount of time, so you certainly don't want to do that more than once. Also if the tide tables are going to be changed each year then keeping them in a separate file minimises what has to be changed.

We would then also have a realtime trigger that fires every minute and runs the script:

```
-- check if our index variable is initialised
if not h then
    h = 0
end

-- make sure we don't run off the end of the table
while high_tides[h*5] do
    -- compare the current time against table
    year = realtime.year
    if high_tides[h*5] > year then
        -- not yet reached this entry
        return
    elseif high_tides[h*5] == year then
        month = realtime.month
        if high_tides[(h*5)+1] > month then
            -- not yet reached this entry
            return
        elseif high_tides[(h*5)+1] == month then
            day = realtime.monthday
            if high_tides[(h*5)+2] > day then
                -- not yet reached this entry
                return
            elseif high_tides[(h*5)+2] == day then
```

```
                    hour = realtime.hour
                    if high_tides[(h*5)+3] > hour then
                        -- not yet reached this entry
                        return
                    elseif high_tides[(h*5)+3] == hour then
                        minute = realtime.minute
                        if high_tides[(h*5)+4] > minute then
                            -- not yet reached this entry
                            return
                        elseif high_tides[(h*5)+4] == minute then
                            -- found a match
                            start_timeline(1)
                                h = h+1
                                return
                        end
                    end
                end
            end
        end
        -- not yet reached this entryController
        h = h+1
    end
```

This may look rather dense and scary - but on closer inspection you'll see that it is just a lot of if tests nested inside each other to check each part of the date and time against the data table. When we detect that the current time is later than the current entry in the table then we move on to looking at the next entry.

Implementing an interactive game for a Science Museum

In an exhibit children are posed questions and have to select answers from an array of numbered buttons. The buttons are large with RGB backlights that are controlled by a Controller to highlight choices and indicate right and wrong answers. Questions are displayed by a slide projector which is under RS232 control from the Controller. The buttons are wired to contact closures on the Controller and on RIOs, so that the Controller can check answers and determine the progress of the game accordingly. The lighting in the rest of the room is designed to mimic a popular TV quiz show to retain the children's interest, with different timelines for each stage of the game.

I am not going to work through this example - but the key point is that it should now be clear to you that a Controller could be used to implement this sort of advanced interactive exhibit with the use of script. Try breaking down the problem into discrete parts and you will find that no individual part of this is difficult - although getting it all to function together reliably would no doubt require a lot of work. The Controller is a viable alternative to custom software running on a PC and has clear advantages in terms of durability and cost.

More information

In this document we have only covered the basic concepts that are needed to understand or write useful scripts for the Controllers. For more extensive information on the Lua language there are two documents, both of which are available online at http://www.lua.org or can be bought as books from Amazon.

- Lua 5.0 Reference Manual
- Programming in Lua

Note that Unison Mosaic currently uses Lua 5.0 rather than the newer Lua 5.1, although the differences are irrelevant for the kind of scripts we are focusing on here.

Custom Preset Programming Guide

Custom Presets use a Lua script to define an effect that can be played back on a Matrix. You can use this to create effects that are not available as standard in Designer. Custom Presets are managed using the Media window.

Basics

Custom presets use Lua scripts to define an animation.

For each pixel (x,y) of each frame of that animation, a `pixel` function is called which returns three numbers, between 0 and 255, which represent the red, green and blue components of the colour of that pixel. Pixel (0,0) is in the top left of the frame, with the positive x axis pointing right and the positive y axis pointing down.

Here is the most simple example of a custom preset:

Listing 1

```
function pixel(frame,x,y)
    return 255,0,0
end
```

This fills every pixel of every frame with red. If you do not return all three components of the pixel's colour, the missing components are assumed to be 0, so the following function is equivalent to Listing 1:

Listing 2

```
function pixel(frame,x,y)
    return 255
end
```

A real example

To demonstrate what can be achieved with custom presets, we are going to build up a real example as concepts are introduced throughout this guide.

To start, we are going to create a preset that renders a series of vertical red bands:

Listing 3

```
-- width of the bands in pixels
band_width = 4
-- space between bands in pixels
band_spacing = 1
-- modulo operator (a%b)
function mod(a,b)
    return a - math.floor(a/b)*b
end

-- the pixel function
function pixel(frame,x,y)
    -- use the modulo operator to split the horizontal axis into bands
```

```
and
    -- decide if we are in the band or in the separator between bands
    if (mod(x,band_width+band_spacing)<band_width) then
        -- in band
        return 255,0,0
    else
        -- in band separator
        return 0,0,0
    end
end
```

You will note that we have defined a new function, `mod`, to implement the modulo operator. This was done to make the script more readable. We will discuss user-defined functions again later.

We also defined two variables, `band_width` and `band_spacing`. These we placed outside of the `pixel` function because they are the same for every pixel of every frame of the effect, so it is more efficient to not execute the assignment for every pixel. Any code outside of the pixel function is executed once, before the `pixel` function is called for the first time.

Animation

Filling every frame of an animation with a single colour is not very exciting, so we can use the `frame` argument to change the colour of a given pixel (x,y) based on the current frame.

Here is an example:

Listing 4

```
function pixel(frame,x,y)
    if (x<frame) then
        return 255,0,0
    else
        return 0,0,0
    end
end
```

This creates a red horizontal wipe, advancing 1 pixel towards the right for each frame. You may have noted that once the wipe reaches the right side of the frame, the whole frame stays red for a period of time before the animation loops back to the beginning. This is because the number of frames exceeded the number of pixels across the frame.

Ideally, we want our effects to loop seamlessly. To do this, we introduce three global variables that have been already been defined for you:

- `frames` - the total number of frames in the animation
- `width` - the width of the animation in pixels
- `height` - the height of the animation in pixels

We can rewrite Listing 4 as follows:

Listing 5

```
function pixel(frame,x,y)
    -- calculate the progress through the animation
    local t = frame/frames
    -- compare the fraction across the effect with the animation pro-
gress
    if (x/width<t) then
        return 255,0,0
    else
        return 0,0,0
    end
end
```

Now, once the red wipe reaches the right side of the frame, it immediately jumps back to the start. Returning to our vertical band example, we are going to introduce animation by changing the height of each band over time:

Listing 6

```
-- width of the bands in pixels
band_width = 4
-- space between bands in pixels
band_spacing = 1

-- get the combined width of band and separator
local total_band_width = band_width+band_spacing
-- get the number of visible bands
local bands = width/total_band_width

-- modulo operator (a%b)
function mod(a,b)
    return a - math.floor(a/b)*b
end

-- the pixel function
function pixel(frame,x,y)

    if (mod(x,total_band_width)>=band_width) then
        -- in band separator
        return 0,0,0
    end

    -- get the band in which this pixel falls
    local band = math.floor(x/total_band_width)

    -- get the fraction through the effect
    local t = frame/frames

    -- get the height of the band in which this pixel falls
    local band_height = (math.sin((band/bands+t)*math.pi*2)+1)/2
```

```
        -- adjust y to be relative to the center of the effect
        y = y-(height/2)+0.5

        -- decide if this pixel is inside the band
        if (math.abs(y)/(height/2) <= band_height) then
            return 255,0,0
        else
            return 0,0,0
        end
    end
```

We are using a sine function to set the height of each band, where the argument to the sine function is offset based on the index of the band and the current fraction through the effect. The result of this is that the height of each band differs from its neighbour according the sine function, and this relationship is modified over time to create a ripple.

More colours than just red

So far, we have just been creating red effects, but there are more colours than red, so why should we stick with that? We will modify the vertical band example to show how different colours can be created. For this example, we introduce the built-in function, `hsi_to_rgb`, which converts an HSI (hue, saturation, intensity) colour into an RGB (red, green, blue) colour:

Listing 7

```
    -- width of the bands in pixels
    band_width = 4
    -- space between bands in pixels
    band_spacing = 1

    -- get the combined width of band and separator
    local total_band_width = band_width+band_spacing
    -- get the number of visible bands
    local bands = width/total_band_width

    -- modulo operator (a%b)
    function mod(a,b)
        return a - math.floor(a/b)*b
    end

    -- rainbow lookup
    function rainbow(hue)
        return hsi_to_rgb(hue*math.pi*2,1,1)
    end

    -- the pixel function
    function pixel(frame,x,y)

        if (mod(x,total_band_width)>=band_width) then
            -- in band separator
            return 0,0,0
        end
        -- get the band in which this pixel falls
        local band = math.floor(x/total_band_width)

        -- get the fraction through the effect
        local t = frame/frames

        -- get the height of the band in which this pixel falls
        local band_height = (math.sin((band/bands+t)*math.pi*2)+1)/2

        -- adjust y to be relative to the center of the effect
        y = y-(height/2)+0.5

        -- decide if this pixel is inside the band
        local h = math.abs(y)/(height/2)
        if (h <= band_height) then
            return rainbow(band/bands+t)
        else
            -- offset hue by quarter
            return rainbow((band/bands+t)+0.25)
        end
    end
```

We have defined a new function, `rainbow`, which returns a fully saturated r,g,b value for a given hue. This function is then called with different arguments depending on whether on not a pixel falls inside or outside of a band.

User-defined functions can be used whenever you want to use a similar piece of code in multiple places with differing arguments.

Running this script, you will see that the bands are now coloured with a rainbow which changes over time, and the area above and below the band is filled with a colour that is pi/2 radians out of phase with the band's colour.

Working with colours

Working with colours as 3 separate components can produce a wide variety of effects, but sometimes it is more convenient to treat a colour as a single entity. We can do that with the colour library.

To create a variable of type colour, call `colour.new()`, passing in three values between 0 and 255 which represent the red, green and blue components of the colour, i.e:

```
local c = colour.new(255,0,0)
```

The variable `c` has the type colour and represents red. Colours have three properties, `red`, `green` and `blue`, which can be used to access and alter that colour. Here is a simple example using the colour type:

Listing 8

```
    function pixel(frame,x,y)
        local c = colour.new(255,0,0)
        return c.red,c.green,c.blue
    end
```

This fills every pixel of every frame with red.

Earlier in this document, we stated that the `pixel` function should return 3 numbers, representing the red, green and blue components of a colour. This was not the entire truth. We are also allowed to return a single variable of type colour. This function is therefore equivalent to Listing 8:

Listing 9

```
function pixel(frame,x,y)
    local c = colour.new(255,0,0)
    return c
end
```

Once again, we return to our vertical band example and use colour variables to specify the band colour and the background colour:

Listing 10

```
-- width of the bands in pixels
band_width = 4
-- space between bands in pixels
band_spacing = 1
-- the colour of the band
band_colour = colour.new(255,0,0)
-- the colour of the space between bands
background_colour = colour.new(0,0,255)

-- get the combined width of band and separator
local total_band_width = band_width+band_spacing
-- get the number of visible bands
local bands = width/total_band_width

-- modulo operator (a%b)
function mod(a,b)
    return a - math.floor(a/b)*b
end

-- the pixel function
function pixel(frame,x,y)

    if (mod(x,total_band_width)>=band_width) then
        -- in band separator
        return background_colour
    end

    -- get the band in which this pixel falls
    local band = math.floor(x/total_band_width)

    -- get the fraction through the effect
    local t = frame/frames

    -- get the height of the band in which this pixel falls
    local band_height = (math.sin((band/bands+t)*math.pi*2)+1)/4
```

```
        -- adjust y to be relative to the center of the effect
        y = y-(height/2)+0.5

        -- decide if this pixel is inside the band
        if (math.abs(y)/height<=band_height) then
            return band_colour
        else
            return background_colour
        end
    end
```

We have added two variables, `band_colour` (red) and `background_colour` (blue) and are now returning those values rather than the r,g,b values that we were using previously. You should now see red bands rippling over a blue background.

A simple gradient

The colour library also includes an `interpolate` function, which takes two colours and a fraction and returns a new colour that is linearly interpolated between the two colours. For example:

Listing 11

```
    local red = colour.new(255,0,0)
    local blue = colour.new(0,0,255)

    function pixel(frame,x,y)
        -- interpolate between red and blue using the horizontal dis-
    placement of x
        -- note that we use (width-1) so the rightmost pixel is completely
    blue
        return colour.interpolate(red,blue,x/(width-1))
    end
```

This creates a horizontal red to blue gradient. We could have created the same gradient without the colour library as follows:

Listing 12

```
    function pixel(frame,x,y)
        local f = x/(width-1)
        return 255*(1-f),0,(255*f)
    end
```

However, if you changed your mind about the colours that you wanted for your gradient, it would be significantly harder to alter Listing 12 than it would be to change the colours in the first two lines of Listing 11.

Working with gradients

The gradient library adds support for more complicated gradients that cannot be achieved by interpolating between two colours.

To create a new variable of type gradient, call `gradient.new()`, passing in two colours, i.e:

```
local c1 = colour.new(255,0,0)
local c2 = colour.new(0,0,255)
local g = gradient.new(c1, c2)
```

To find the colour of the gradient at a specific point, use the `lookup` function, passing in a number between 0 and 1. For example:

Listing 13

```
local red = colour.new(255,0,0)
local blue = colour.new(0,0,255)
local g = gradient.new(red, blue)

function pixel(frame,x,y)
    -- note the use of the colon operator
    return g:lookup(x/(width-1))
end
```

This creates a horizontal gradient from red to blue, but we have already seen that there are other ways to generate the same result which will probably be more efficient. To show where the gradient library offers more power:

Listing 14

```
local red = colour.new(255,0,0)
local blue = colour.new(0,0,255)
local g = gradient.new(red,blue)

-- add a third point to the middle of the gradient
local green = colour.new(0,255,0)
g:add_point(0.5,green)

function pixel(frame,x,y)
    return g:lookup(x/(width-1))
end
```

We used the `add_point` function to insert a green colour midway between the red and the blue colours. This generates a horizontal gradient that fades from red to green to blue.

Back to the vertical band example, we will use a gradient to colour the bands:

Listing 15

```
-- width of the bands in pixels
band_width = 4
-- space between bands in pixels
band_spacing = 1
-- the colour of the band
band_gradient = gradient.new(colour.new(255,0,0), colour.new
(255,255,0))
-- the colour of the space between bands
background_colour = colour.new(0,0,0)
```

```
-- get the combined width of band and separator
local total_band_width = band_width+band_spacing
-- get the number of visible bands
local bands = width/total_band_width

-- modulo operator (a%b)
function mod(a,b)
    return a - math.floor(a/b)*b
end

-- the pixel function
function pixel(frame,x,y)

    if (mod(x,total_band_width)>=band_width) then
        -- in band separator
        return background_colour
    end

    -- get the band in which this pixel falls
    local band = math.floor(x/total_band_width)

    -- get the fraction through the effect
    local t = frame/frames

    -- get the height of the band in which this pixel falls
    local band_height = (math.sin((band/bands+t)*math.pi*2)+1)/2

    -- adjust y to be relative to the center of the effect
    y = y-(height/2)+0.5

    -- decide if this pixel is inside the band
    local h = math.abs(y)/(height/2)
    if (h<=band_height) then
        return band_gradient:lookup(h)
    else
        return background_colour
    end
end
```

The `band_gradient` variable is initialised as a red to yellow gradient, and we use `band_gradient:lookup (h)` to determine the colour of the band at height h.

Working with properties

Custom presets can have properties which will be exposed in Designer whenever the preset is placed on a timeline. This allows a single custom preset to create a wide variety of effects. It also means that you do not have to create near-identical copies of custom presets just to change one parameter, for example, a colour. You can just expose a colour property and specify the desired colour when the preset is placed on a timeline.

To define a property, you would call the function:

```
property(name, type, default_value, ...)
```

This must be added to your script outside of any function call.

`name` is a string and must be unique within a custom preset and must not contain spaces. This name will be used as the name of a global variable that is available in your script, whose value will depend on what has been set for a given instance of your custom preset.

`type` is the type of the property. It can be one of the following values: `BOOLEAN`, `INTEGER`, `FLOAT`, `COLOUR` and `GRADIENT`. This determines what sort of control is presented to the user when placing a custom preset on a timeline.

`default_value` is the initial value of a property when first added to a timeline. The value passed in here depends on the type of the property, and this is outlined below.

Certain types of properties also allow some addition arguments to be specified, and these will also be described for each type below:

Boolean properties

```
property("invert", BOOLEAN, true)
```

The default value should be `true` or `false`.

Integer properties

```
property("count", INTEGER, number, [min], [max], [step])
```

The default value should be a number between `min` and `max`.

- `min` is the minimum allowed value (default: -2147483648)
- `max` is the maximum allowed value (default: 2147483647)
- `step` is the difference between allowed values (default: 1)

`min`, `max` and `step` are optional.

Float properties

```
property("count", FLOAT, number, [min], [max], [resolution])
```

The default value should be a number between `min` and `max`.

- `min` is the minimum allowed value
- `max` is the maximum allowed value
- `resolution` is the number of decimal places to display (default: 2)

`min`, `max` and `resolution` are optional.

Colour properties

```
property("background", COLOUR, red, green, blue)
```

`red`, `green` and `blue` are the default values of the components of the colour.

Gradient properties

```
property("gradient", GRADIENT, {fraction, red, green, blue}, ...)
```

The default value of a gradient is a list of fractions and colours, where `fraction` is in the range [0-1] and specifies where in the gradient the colour is, and `red`, `green` and `blue` is the colour at that position and are in the range [0-255]. You can specify multiple points. For example:

```
property("gradient", GRADIENT, 0.0, 255, 0, 0, 1.0, 0, 0, 255)
```

creates a red (255,0,0) point at the start (0.0) and a blue ((0,0,255) point at the end (1.0).

To demonstrate a real example of using properties in scripts:

Listing 16

```
property("g", GRADIENT, 0.0, 255, 0, 0, 1.0, 0, 0, 255)

function pixel(frame,x,y)
    return g:lookup(x/(width-1))
end
```

This, by default, creates a horizontal gradient from red to blue, as we saw in Listing 13. However, when this preset is placed on a timeline, there will be a gradient editor available, and you will be able to alter the gradient to be any colour you wish, without having to recompile the script or having to duplicate the custom preset with some small alterations.

We will now modify our vertical band example to expose some properties to make a very versatile effect:

Listing 17

```
-- width of the bands in pixels
property("band_width", INTEGER, 4, 1)
-- space between bands in pixels
property("band_spacing", INTEGER, 1, 0)
-- the wavelength of the ripple (in terms of current width)
property("wavelength", FLOAT, 1, 0, 16, 2)
-- the direction of the ripple
property("reverse", BOOLEAN, false)
-- the colour of the band
property("band_gradient", GRADIENT, 0, 255, 0, 0, 1, 255, 255, 0)
-- the colour of the space between bands
property("background_colour", COLOUR, 0, 0, 0)

-- get the combined width of band and separator
local total_band_width = band_width+band_spacing
-- get the number of visible bands
local bands = width/total_band_width

-- modulo operator (a%b)
function mod(a,b)
    return a - math.floor(a/b)*b
end

-- the pixel function
function pixel(frame,x,y)
```

```
        if (mod(x,total_band_width)>=band_width) then
            -- in band separator
            return background_colour
        end

        -- get the band in which this pixel falls
        local band = math.floor(x/total_band_width)

        -- get the fraction through the effect
        local t = frame/frames

        -- optionally reverse the ripple
        if (reverse) then t = -t end

        -- get the height of the band in which this pixel falls
        local band_height = (math.sin((band/bands/wavelength+t)*math.pi*2)
    +1)/2

        -- adjust y to be relative to the center of the effect
        y = y-(height/2)+0.5

        -- decide if this pixel is inside the band
        local h = math.abs(y)/(height/2)
        if (h<=band_height) then
            return band_gradient:lookup(h)
        else
            return background_colour
        end
    end
```

You will notice that adding properties to the example involved little more than changing the variable definitions at the start of the script. There are also two new properties, `wavelength`, for setting the wavelength of the ripple, and `reverse`, for changing the direction of the ripple.

By adjusting the values of the properties, we can now create a variety of different effects without having to alter the script again.

Colour library summary

`colour.new(r,g,b)`

Returns a new colour that represents the RGB color specified by the components `r`, `g` and `b`. `r`, `g` and `b` will be limited to the range [0,255].

`colour.interpolate(c1,c2,f)`

Returns the colour that is linearly interpolated between colour `c1` and colour `c2` at fraction `f`. `f` can fall outside of the range [0,1] and the returned colour will be extrapolated accordingly.

Properties

`c:red`

The value of the red component [0-255] of colour `c`.

`c:green`

The value of the green component [0-255] of colour `c`.

`c:blue`

The value of the blue component [0-255] of colour `c`.

Gradient library summary

`gradient.new(c1,c2)`

Returns a new gradient with colour `c1` at the start and colour `c2` at the end.

Functions

`g:lookup(f)`

Returns the colour at fraction `f` through the gradient `g`. `f` will be limited to the range [0,1].

`g:add_point(f, c)`

Adds the colour `c` to the gradient `g` at fraction `f`.

Built-in functions

`dist(x1,y1,x2,y2)`

Returns the distance between coordinate (`x1`,`y1`) and coordinate (`x2`,`y2`)

`dist_from_center(x,y)`

Returns the distance between coordinate (`x`,`y`) and the center of the frame. This is not the same as calling `dist(x,y,width/2,height/2)`. It takes into account the fact that the center of the frame may fall in the middle of a pixel. For example, if `width` and `height` were equal to 5, the center of the frame is the center of the pixel at coordinate (2,2), but calling `dist(2,2,width/2,height/2)` will return 0.707, which is the distance between the top left of pixel (2,2) and its center. Calling `dist_from_center(2,2)`, where `width` and `height` are equal to 5, will return 0.

`print(message)`

Prints `message` in the debugger's Output window.

You are advised to remove calls to this function when you have finished debugging because it will allow the script to run faster when used in programming.

`rgb_to_hsi(red,green,blue)`

Converts an RGB (red, green, blue) colour to an HSI (hue, saturation, intensity) colour. `red`, `green` and `blue` are in the range [0-255]. Returns three numbers, hue is in [0-2PI] radian, saturation and intensity are in the range [0-1].

`hsi_to_rgb(hue,saturation,intensity)`

Converts an HSI (hue, saturation, intensity) colour into an RGB (red, green, blue) colour. `hue` is in [0-2PI] radians, `saturation` and `intensity` are in the range [0-1]. Returns three numbers in the range [0-255].

TPC learning infrared receiver

The TPC may be taught to recognise up to 16 different infrared (IR) codes from a standard IR remote control. When a key on the remote control is pressed during normal operation, the TPC will react as though one of its user interface controls has been touched.

The TPC does not have to be part of a networked system to learn IR codes, all that is required is PoE power and the donor remote control:

To enter Learn Mode:

1. Enter by pressing the CFG (config) button. This is located underneath the magnetic overlay at the top left of the display, underneath the Reset button.

- The screen will display the IR configuration interface.

To learn an IR code:

1. Press the Set button alongside the code to be learnt.

- A progress indication will appear on the left of the row.

2. Within ten seconds, point the IR remote at the TPC and press the desired key.

- The progress indication will be replaced with a tick icon when the code has been learnt.

To test an IR slot

1. Point the IR remote at the TPC and press a key.

- If the IR code received is associated with an IR slot, the slot will be highlighted.

2. Release the key on the IR remote.

- The IR slot will no longer be highlighted.

To erase an IR code:

1. Press the Clear button alongside the code to be erased.

- The tick icon next to the code will disappear.

To exit Learn Mode:

1. Press the CFG (config) button.

- The screen will display the user interface for the loaded presentation, or indicate that no user interface is present on the memory card.

BPS learning infrared receiver

The BPS may be taught to recognise up to 8 different infrared (IR) codes from a standard IR remote control. When a key on the remote control is pressed during normal operation, the BPS will react as though one of its 8 buttons has been pressed.

The BPS does not have to be part of a networked system to learn IR codes, all that is required is PoE power and the donor remote control:

To enter Learn Mode:

1. Enter by holding down the bottom two buttons while pressing and releasing reset.
   - The buttons will display a clockwise chase sequence
2. Release the bottom two buttons.
   - Each button will flash quickly (4Hz) if an IR code has been learnt, or slowly (1Hz) if not
   - No network communication will operate while in Learn Mode
   - Learn Mode will automatically exit after 60 seconds of inactivity

To learn an IR code:

1. Briefly press and release a single button which should learn the IR code.
   - The button will start flashing rapidly (8Hz) and the other buttons will extinguish
2. Within ten seconds, point the IR remote at the BPS and press and hold the desired key.
   - The buttons will display a clockwise chase sequence when the IR code has been learnt
3. Release the key on the IR remote.
   - The button now will be flashing quickly (4Hz) to indicate that it has an IR code stored

To erase an IR code:

1. Press and hold for three seconds the button which should erase its IR code.
   - The buttons will display a clockwise chase sequence when the IR code has been erased
2. Release the button.
   - The button will now be flashing slowly (1Hz) to indicate that it has no IR code stored

To test an IR code:

1. Point the IR remote at the BPS and press and hold the key to test.
   - The button(s) that has learnt this code will illuminate solidly, all others will extinguish
2. Release the key and test the others.

To exit Learn Mode:

1. Press the reset button or wait for 60 seconds.

- The buttons will now revert to normal operation
- Network communication will resume

File formats

Fixture plan file format (*.csv)

The fixture plan file format is versioned to ensure that CSV files generated for Designer 1.7.x and later will be imported into any version of Designer from 1.7.0 onwards. Version information must be added to the first line of the CSV file as follows:

#version=2

The current version is 2.

Lines starting with # are comments and are ignored (except the version line). Blank lines are ignored.

Fixture type syntax

@[manufacturer id],[model id](,[mode id])

where [manufacturer id] and [model id] are the numbers which can be found in the fixture's identification fields in Setup. [mode id] is optional, leaving it out assumes mode 0 or that it is irrelevant.

- The fixture type must be specified before any fixture position line(s)
- Multiple fixture type lines can appear consecutively in a file for batches of different fixture types

Fixture position syntax

[fixture number],[name],[x position],[y position],[rotation angle],[width],[height],[comment 1],[comment 2]

where [fixture number], [x position], [y position], [rotation angle], [width] and [height] are mandatory numbers and [name], [comment 1] and [comment 2] are strings which may be left blank. [rotation angle] is specified in degrees, clockwise from vertical.

- Fixture position lines must be specified after a fixture type line
- Fixture numbers must be unique in the file and must not already exist in the project
- The plan will be resized to accommodate the fixture positions, after getting confirmation that it is alright to do so

Example

```
# LEDs
@0,5
1,,0,0,0,50,50
2,,30,0,0,50,50
3,,60,0,0,50,50
4,,90,0,0,50,50

# Dimmers
@0,0
100,,100,100,0,25,25
```

```
101,,125,100,0,25,25
102,,100,125,0,25,25
103,,125,125,0,25,25
```

Pixel matrix file format (*.csv)

The pixel matrix file format is versioned to ensure that CSV files generated for Designer 1.9.x and later will be imported into any version of Designer from 1.9.0 onwards. Version information must be added to the first line of the CSV file as follows:

```
#version=1
```

The current version is 1.

Lines starting with # are comments and are ignored (except the version line). Blank lines are ignored.

Fixture position syntax

[fixture number](.[fixture element index]),[x position],[y position],[rotation angle]

where [fixture number], [fixture element index], [x position] and [y position] are numbers. [rotation angle] is optional (not valid for single-element fixtures) and is specified in degrees, clockwise from vertical. The fixture element index is only required if the fixture specified by fixture number is a compound fixture.

NOTE: On a matrix, a fixture's position is where the first element of the unrotated fixture lies. Fixtures are rotated about their center. Therefore a rotated fixture will appear to have a different position to that given in the imported file.

Example

```
# Simple matrix
1,0,0
2,30,0
3,0,30
4,30,30
5.1,0,60
5.2,30,60
```

Web server query interface

The query interface of a Controller provides access to timeline status information, text slot contents, etc. for custom web pages.

Syntax

If you `GET` the URL:

```
http://xxx.xxx.xxx.xxx/query/command[?args]
```

and *`command`* and *`args`* are recognised, the controller will return the result of the query, else, it will return a 204 (No Response) HTTP header.

Data will be returned as XML. Recognised commands are:

- `system`
- `currentTime`
- `timeline`
- `timelineStatus`
- `textSlot`
- `variable`

Commands

Get Controller Settings

Get a summary of the Controller's settings.

- Command: `system`
- Args: `none`

Sample Response

```
<response>
  <hardwareType>LPC2</hardwareType>
  <serialNumber>002056</serialNumber>
  <firmwareVersion>
    <major>1</major>
    <minor>7</minor>
    <point>133</point>
  </firmwareVersion>
  <networkInterface>
    <ipAddress>172.20.1.9</ipAddress>
    <subnetMask>255.255.0.0</subnetMask>
    <gateway>172.20.0.250</gateway>
  </networkInterface>
  <cfCardSizeKb>1024</cfCardSizeKb>
  <date>2010-04-23</date>
  <time>08:59:14</time>
```

```
    <sunrise>06:55:00</sunrise>
    <sunset>19:02:00</sunset>
    <projectName>My First Show</projectName>
    <projectAuthor>Joe Bloggs</projectAuthor>
    <projectUuid>00000000-0000-0000-0000-000000000000</projectUuid>
    <projectUploadDate>2010-04-21T13:36:18</projectUploadDate>
    <controllerNumber>1</controllerNumber>
  </response>
```

Notes

`<date>`, `<time>`, `<sunrise>` and `<sunset>` are in local time.

MSC X will have a `<networkInterface2>` tag with the network details of the second interface.

AVC will not have a `<projectUploadDate>` tag.

Get Current Time

Get a time and date from the Controller's system clock in local time.

- Command: `currentTime`
- Args: `none`

Sample Response

```
  <response>
    <currentTime>
      <date>2002-05-30</date>
      <time>09:00:00</time>
    </currentTime>
  </response>
```

Get Timelines

Get a list of timelines in the project. `id` is the timeline number.

- Command: `timeline`
- Args: `id=1 / id=1,3,5 / id=1-5 / id=1-3,5,8-10 (optional, defaults to ALL)`

Sample Response

```
  <response>
    <timeline id = "1">
      <name>My First Timeline</name>
      <length>P00H30M00.00S</length>
      <timeSource>
        <type>Timecode</type>
        <bus>2</bus>
    </timeSource>
    <timeOffset>P00H00M00.00S</timeOffset>
    <timeFormat>SMPTE30</timeFormat>
  </timeline>
  <timeline id = "3">
    ...
  </timeline>
</response>
```

Get Timeline Statuses

Gets the current status of timelines in the project. id is the timeline number.

- Command: `timelineStatus`
- Args: `id=1 / id=1,3,5 / id=1-5 / id=1-3,5,8-10 (optional, defaults to ALL)`

Sample Response

```
<response>
  <timelineStatus id = "1">
    <playState>Running</playState>
    <position>P00H14M03.50S</position>
    <onStage>true</onStage>
  </timelineStatus>
  <timelineStatus id = "3">
    <playState>Released</playState>
  </timelineStatus>
</response>
```

Notes

Valid values for `<playState>` are:

- `Running`
- `Halted`
- `Held at end`
- `Released`
- `Idle`

Get Text Slots

Get a list of text slots in the project, along with their current values.

- Command: `textSlot`
- Args: `slot_name (optional, defaults to ALL)`

Sample Response

```
<response>
```

```
  <textSlot name = "slot_name">slot value</textSlot>
</response>
```

Notes

Multiple text slots can be requested by adding multiple query arguments, for example:

```
GET http://xxx.xxx.xxx.xxx/query/textSlot?slot1&slot2&slot3
```

If a slot name is not recognised, it will be ignored.

Get Global Lua Variables

Get the value of a global Lua variable.

- Command: `variable`
- Args: `variable_name (required)`

Sample Response

```
<response>
  <variable name = "variable_name">variable value</variable>
</response>
```

Notes

Multiple variables can be requested by adding multiple query arguments, for example:

```
GET http://xxx.xxx.xxx.xxx/query/vari-
able?variable1&variable2&variable3
```

If a variable name is not recognised, or if the variable's value cannot be represented as text, the `<variable>` tag will contain no value.

Example

An example file can be found here.

Glossary

B

bootloader

Bootstrap loader; a small software program, stored in internal flash memory, that is responsible for loading the firmware or operating system.

C

compound fixture

A lighting fixture containing more than one controllable element, for example an LED batten consisting of a number of identical elements or pixels.

D

DALI

"Digital Addressable Lighting Interface"; an industry standard digital lighting control protocol.

DHCP

"Dynamic Host Configuration Protocol"; a method of automatically assigning IP addresses.

DMX

USITT DMX512; an industry standard digital lighting control protocol.

E

eDMX

A shorthand term for DMX-over-Ethernet protocols, for example Art-Net II.

F

firmware

The embedded operating system, stored in internal flash memory or on the Compact Flash card.

fixture

Lighting instrument or luminaire.

G

group

A collection of fixtures or elements (pixels) within a fixture that provide a very useful shortcut for selecting and programming them together as one.

I

IP address

"Internet Protocol" address, in the form xxx.xxx.xxx.xxx, which specifies the unique address for networked equipment.

M

matrix

A two-dimensional array of fixtures such that each fixture, or element within a compound fixture, is mapped to a pixel of the array.

MIDI

"Musical Instrument Digital Interface"; an industry standard communications protocol for musical instruments.

mover

Any fixture that has control parameters beyond colour mixing (RGB, CMY etc) and intensity, typically an automated light.

N

NTP

"Network Time Protocol"; a means of transmitting time signals over a computer network, used to set realtime clocks automatically to the correct time.

P

preset

The basic building block that is placed on a timeline to define what a fixture, or group of fixtures, is to do. Roughly analogous to a cue.

R

RDM

"Remote Device Management"; an extension to the USITT DMX512 protocol that allows for bi-directional communication with the fixture for remote configuration and diagnostics purposes.

RS232

EIA-232; an industry standard communications protocol for computing and telecommunications equipment.

RS485

EIA-485; an industry standard communications protocol for computing and industrial equipment.

T

timeline

The framework used to determine which presets are applied to which fixtures, when and for how long. Roughly analogous to a cuelist.

V

variable

A value that can be captured from an input string that is used to determine the outcome of an action.

W

watchdog

A hardware device that monitors a microprocessor and automatically forces a reset if the microprocessor stops responding.

wildcard

A method of specifying which character(s) of an input string should be ignored as padding. Wildcards are also captured as variables and can be considered such if used to determine the outcome of an action.