

# LSC's Clarity

reviewed by Rob Halliday

Now here's a funny thing: the traditional complaint from lighting manufacturers is that there is no money to be made from lighting consoles. Given the laws of capitalism, you'd therefore expect fewer and fewer consoles to be available. In fact, this seems to be a market segment bursting with activity, a constant flow of new products. The 'big players' attract the most attention - think ETC, think MA. But there are a whole host of other manufacturers, some now looking to expand their reach from their home countries . . .



LSC Lighting Systems, based near Melbourne in Australia, falls into that category. The company has been building lighting control products for almost 30 years; its current range includes a wide range of dimming and power distribution products plus three console lines: Minim (a 24-channel, two-preset plus memory fader board), Xtc (a hands-on controller for up to 100 fixtures over two DMX universes), and maXim (a memory-plus-fader console aimed at live control or memory playback).

Now they've added a fourth: Clarity, designed for running more complex rigs, particularly those involving LED arrays, media servers and the like.

Clarity departs from the existing LSC model in one important way: it is not a lighting console *per se*. Instead, it's a software-based lighting package that can run on Windows PCs or Apple Macs, plus a range of wings that can be added if more hands-on control is required.

This is not a new concept, various others having offered it as a 'value for money' solution over the years and it having gained popularity more recently through the Hog iPC range. 'Serious' users have tended to stick to 'real' consoles, though - but as a younger generation increasingly occupies the hot seat behind lighting controllers, maybe the time is coming for the PC+wings approach. Perhaps LSC's timing will turn out to be prescient.

## Software

Let's take a look at the software first, since the premise is that you can run Clarity with nothing more than a computer of some kind: LSC specify a Windows XP, Vista or 7-running Pentium 4 PC with 512Mb of RAM, 300Mb of disk space and, ideally, dual monitors, and while they do not yet have a Mac spec, it ran fine both on a 15" 2.8GHz MacBook Pro directly and on the same machine under the VM Fusion emulator (this necessary because, at the time of testing, the Mac version did not support connection to the hardware wings). We looked at Clarity v1.02.

Clarity's start-up screen indicates that it is 'Powered by OpenClear', hinting that this is not solely an LSC project: OpenClear turns out to be the company run by software programmer Nick Denville, one of the Hog 2 development team for a long time and more recently behind the most innovative console of recent history, the time-line controlled Jands Vista. This makes Clarity fascinating: can it combine the strengths of Vista (particularly the ease of creating and editing complex cue timings) while overcoming the frustrations it could cause (notably the difficulty of using it for shows where sequences did not run to pre-definable timing, such as live theatre shows)?

After that splash screen, Clarity lands you in a calm, attractive in shades of grey, yellow and green, single-window control screen with a set of what look like menus but turn out to be tab selectors along the top [Fig. 1: Clarity main window with Open File]. The

single window is actually just a starting point, since many sections of the interface can be opened in separate windows if you have enough screen space.

At first run, you are offered the patch screen, with DMX slots on the right and a long list of fixture manufacturers on the left [Fig. 2: Fixture Library] - this the library from Carallon also used in the Vista, MA2, Eos and others. Clarity does include a fixture editor (currently self-declared as being 'alpha', and so functional but rough around the edges), meaning you can hack together your own fixture on-the-fly if necessary.

Fixture patching is drag-and-drop, as with Vista [Fig. 3: Patch]: pick a light then pull it into the patch, or grab a light already in patch then move it to a new location. You can also specify a 'unit #' - a channel number in conventional speak - and a channel name, if desired. A second tab within patch promises RDM functionality, being able to discover and patch lights on the fly from the console, but we had no RDM-enabled equipment available to test that with. Patching also offers a 'clone from' option [Fig. 4: Clone Fixture], which will clone a new channel or channels from an existing fixture or fixtures in the showfile, attempting to make use of the data in the Carallon library to deal with this intelligently, so that your show programming remains as intact as possible. This makes some intelligent choices about replacement colour and gobos, though because it is not storing real-world values for things like gobo spin speed, these will often be wrong. It's also a shame that you have to do this by cloning to a new fixture; you might just arrive at a venue and want to connect your existing channels to whatever type of light they have. The same intelligence does allow copy-and-paste between fixture types when editing cues, though.

Data output is also controlled from the Patch tab. When using a network connection, *Art-Net* is available. Alternatively,

LSC offers the QX2 and QX4 RDM-DMX nodes, and the VX10 and VX20 playback wings which feature, respectively, two and four RDM-enabled DMX outputs. In every case, Clarity's output capability is actually defined by the licence you purchase, authorised by a USB dongle enabling specific quantities of output slots up to 128 universes. Download Clarity and it will operate as an unlimited version for 30 days, then revert to outputting just 128 DMX slots and a maximum of 32 fixtures.

Rig patched, you switch to the Programmer tab to start making looks [Fig. 5: Programmer Screen]. Clarity offers a set of tabs on the left of the screen, one containing groups of lights then separate tabs for each type of light you have patched. You can pick lights via either tab, but only the latter gives you the level/colour and position indicator graphic for each light [Fig. 6: Groups, Level/Colour Indicators], as well as buttons for selecting all, none, odd or even within the group and for accessing the light's control functions. One final tab, Sort, gives you control over the selection order of your lights [Fig. 7: Sort].

At the bottom of the screen are on-screen controllers for intensity, position, colour and beam, this including thumbnails for gobos in the light [Fig. 8: Parameter Controllers], though gobos across different wheels all get jumbled together. The colour controller takes the form of a colour-picker, but if your light doesn't offer colour mixing, this will be marked with positions corresponding to its fixed colours (this is part of the intelligence that allows information to be cloned from light type to light type, though Clarity does not offer the gel-match library that some Carallon-based



Above: The VX20 (top) and VX10 playback wings.  
Below, left: The QX2 RDM-DMX node.



consoles do). Alternatively, you can control any parameter by click-holding it and moving your mouse [Fig. 9: Dragging Parameters]; this works but is frustrating for only letting you control one thing at a time - particularly problematic with things like framing shutters. New on-screen tools for functions like this are promised. Control- or right-clicking in a parameter offers quick access to capabilities of that parameter (the colours in a colour wheel, for example) plus other things you might want to do - grab into the programmer, clear, fan and so-on [Fig. 10: Ctrl-Click Parameter list]. While Clarity usefully sets parameters like gobos to the gobo name rather than a DMX value, obviating, to some extent, the need to make palettes, as with many consoles it currently doesn't abstract its fixture control as far as it might - if you have two MAC 700s rotating their gobos at the same speed in opposite directions, trying to slow them both down at the same time is not something you'll want to do in front of an audience.

Below the parameter controls are two buttons called 'Programmer 1' and 'Programmer 2' [Fig. 11: Programmer 1, 2]; Clarity is firmly a programmer-based console, so that only changes you make



Fig. 1: Main Window with Open File.

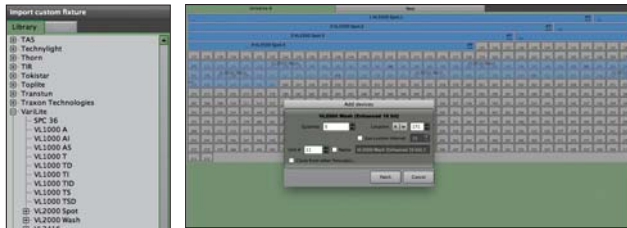


Fig. 3: Patch.

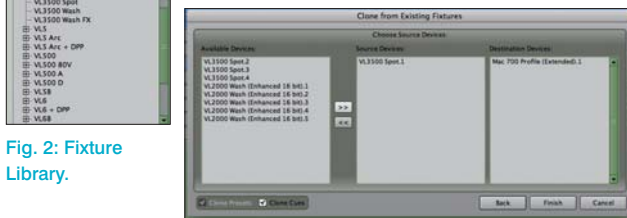


Fig. 4: Clone Fixtures.

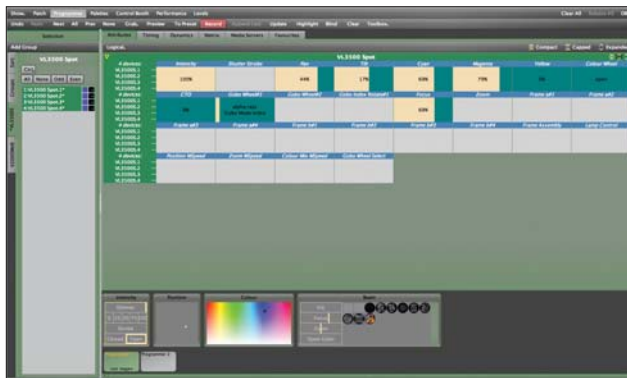


Fig. 5: Programmer Screen.

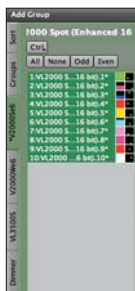


Fig. 6: Groups, Level Colour Indicators.



Fig. 11: Programmer 1, 2.



Fig. 12: Down Centre.



Fig. 7: Sort.



Fig. 8: Parameter Controllers.



Fig. 9: Dragging Parameters.



Fig. 10: Ctrl-Click Parameter List.

are, by default, stored into cues; Clarity's new trick is that there are two programmers you can switch freely between, giving you somewhere to hold temporary channels without recording them into cues.

In the main programmer screen, Clarity tries to live up to its name. If you have 10 lights in position 'Down Centre', many consoles will just show you 'Down Centre' 10 times. Clarity tries to show you that just once, with a graphical bar underneath indicating the numerical value then stored in that preset - this is great for spotting patterns across sets of lights [Fig. 12: Down Centre]. If a group of lights are set to different values, it will try to show you that range: '0 . . . 100%' [Fig. 13: Parameter Range]. You can choose whether to see one line representing this whole block of lights [Fig. 14: Collapsed], a 'minimised' view, or an individual line for each light. It is very nearly brilliant, though something about the range-display logic is not quite right just yet, sometimes showing a range from a palette to itself: 'DC . . . DC'.

Channel control is also where you re-encounter some of the drag-editing options pioneered in Vista. For any group of lights you can hold down the CTRL key to reveal the attribute's three drag handles, allowing you to skew the selection from the top, middle or bottom [Fig. 15: Skewing Parameters]. Similar drag- and skew techniques can be used for setting fades or delays across selections of lights, with Hog-like fade path options also available [Fig. 16: Skewing Times]. Unlike Vista, though, this happens within a traditional cue structure, letting you create complex timings within cues but then have those cues overlap without any problems. As you adjust timings, groupings of lights whose timings are set together are split out as separate blocks on the timing screen, helping you keep see what's doing what; it would be useful if this could be labeled as an aide-memoir, the way people often use part cues [Fig. 17: Timing Blocks]. Setting ranges of things is remarkably easy to achieve, perhaps easy enough even to switch some hard-core command-liners to the graphical interface cause - though for those purists there is a pop-up box for typing times and a command line for setting levels [Fig. 18: Command Line]. Note, though, that Clarity doesn't yet have any understanding of a 'down' time.

Clarity offers conventional palettes for storing data that can be referenced by cues; these are separated into position, colour and beam types but you can store any category to any preset. Unusually, if you do this, Clarity actually presents copies of the same preset in all three categories [Fig. 19: Palettes]. Also unusual, and welcome, is the ability to arrange preset buttons as you like on screen, and the use of a separate 'to preset' button rather than over-burdening the record key. Dynamic presets can be used to store chases made using the dynamic functions, the usual waveforms applied to parameters with the now-familiar drag-and skew-techniques for setting offsets.

Clarity also has a separate palette type called Freesets. These are 'intelligent presets' using the fixture library data - so you can store 'red' from a VL5, and have that same freeset make any other type of light you apply it to turn red, or store a colour range from two lights and then later apply that Freeset to 20 lights to make a rainbow, or store dynamics that can be applied to multiple lights. Freesets are not referenced - when applied they set the selected lights to appropriate numeric values, which you could then store into a traditional preset.

Cues can be stored using Record, which presents the usual options for storing states, changes and for merging or replacing cues and for filtering by parameter family (intensity, position, colour, beam) [Fig. 20: Record Dialogue]. Missing at the moment is an option to record changes cue-only, though you can set a whole cue list to be either tracking or non-tracking, this last making every cue a complete state. The update command allows changes to cues or presets to be stored quickly (though it will not update manually

selected presets, making updating lots of focuses quickly, preset by preset, hard work). Blind editing can be accomplished either by switching one or both programmers to blind, or blind editing the cue as a whole (though this will only show you a cue's changes, not the underlying state; with the level output display currently only showing values not palette names, and the programmer trying to show only your currently selected light rather than all those you've worked with, it is quick to make states in Clarity but can be hard to figure out those states later). You can insert move-in-black cues, but those who like the console to do this work will be disappointed by the lack of automated move-in-black functionality.

Once made, cues can be played back in a variety of ways - in the Control Booth window by dragging-and-dropping a cuelist to create a virtual playback [Fig. 21: Virtual Playback], by dragging and dropping into an on-screen version of an external hardware wing [Fig. 22: Virtual Hardware] (which will also offer the cuelist in the real wing, with its Go, Back and Flash buttons, if connected), or by using Clarity's Performance window [Fig. 23: Performance Window]. This is a really interesting concept for recycling cues in many different forms on the fly for 'busked' shows: an expandable grid of playbacks, into which you can dump any cuelist one or many times, or even just drag dynamic presets applied to groups. You can then over-ride how that version of that cuelist behaves - whether it's a chase or not, its rate, filtering to decide which attributes are played back, even limiting it to a particular sub-range of cues [Fig. 24: Performance Cuelist Options]. These changes do not change the base cuelist, merely how it plays back in that instance, and you can even lock multiple cells to an adjustable master tempo and start or stop whole columns or rows - complex combinations of playbacks - together. It is a great way for making organised mayhem very quickly.

Two other tabs under Clarity's Programmer give access to more complex light types - Matrix and Media Servers. Matrix allows 'pixel grids' to be defined, with some useful tools for defining how real fixtures are patched within the grid [Fig. 25: Matrix]. Still images or movie files (anything supported by QuickTime) can then be dropped onto the grid and scaled, rotated, run or altered in a variety of other ways [Fig. 26: Matrix Control]. The resulting pattern is stored as a PixelSource; each PixelSource appears in the group list as a fixture that can be faded in or out or have its parameters altered on a cue-by-cue basis, just like any other light, with Clarity also offering built-in wipes to make more elegant transitions from look to look. The amount of media functionality packed into what is billed as a lighting console is quite remarkable - Clarity really is more of a mini-media-server, though it is missing the functionality to create looks by just drawing on the array, no media required, that is so appealing in the grandMA.

For situations where more complex media management is required, switch to Clarity's Media Server tab [Fig. 27: Media Servers]. Particularly appealing here is that, while you can set up media servers manually, Clarity also supports the open-standard CFTP/MSEX protocol, plus the Hippotizer's HMAP protocol, for communicating with media servers. This means that, when networked with Hippo v3, ArKaos or Pandoras Box, it can locate media servers on the network, identify them, patch them in the console, then download media thumbnails and display them for easy media selection. If a media server drops off-line, the thumbnails are preserved in Clarity. Though working like this involves a different approach to console setup workflow (since you need the media server present to do the setup), seeing it happen is one of those moments when you feel the technology is just magically on your side.

**Hardware**

So, Clarity's software is a versatile, powerful piece of work - but one which, despite its elegant interface, is ultimately limited by the 'single control point' of a mouse or trackpad. To attempt to overcome this, LSC offers two hardware wings: the VX10 offers 10 playbacks with good-quality faders, go, back and flash buttons plus page-change keys, a blue-backlit LCD display, a grandmaster,

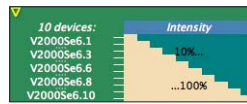


Fig. 13: Parameter Range.



Fig. 14: Collapsed.

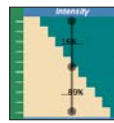


Fig. 15: Skewing Parameters.

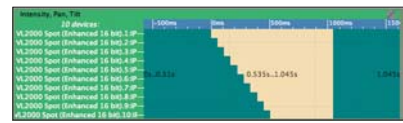


Fig. 16: Skewing Times.

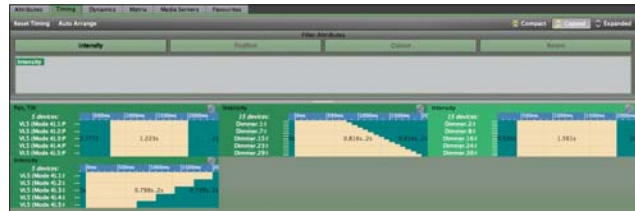


Fig. 17: Timing Blocks.

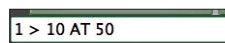


Fig. 18: Command Line.



Fig. 19: Palettes.

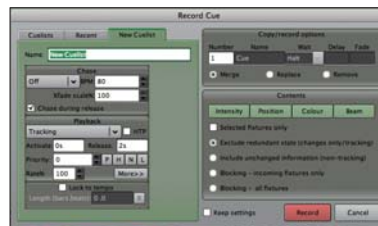


Fig. 20: Record Dialogue.



Fig. 21: Virtual Playback.



Fig. 22: Virtual Hardware.



Fig. 23: Performance Window.



Fig. 24: Performance Cuelist Options.

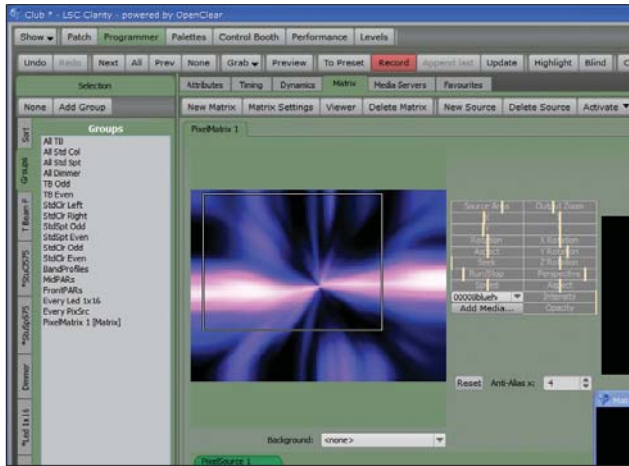


Fig. 25: Matrix.

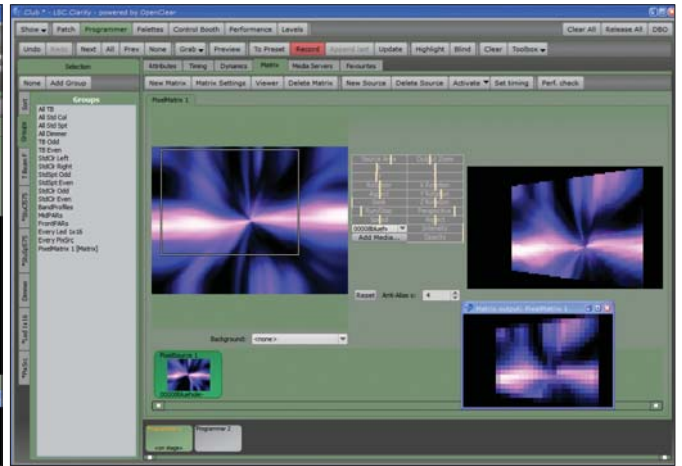


Fig. 26: Matrix Control.

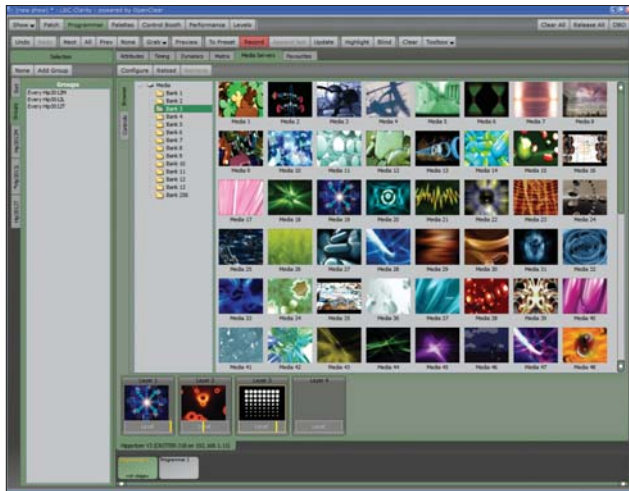


Fig. 27: Media Servers.

master go/back keys and a record button plus two RDM-DMX outputs, a gooseneck lamp connector and a USB socket (with another concealed inside if you want to leave a dongle connected but out of sight). The bigger VX20, which we played with, adds a trackball, parameter type and channel select buttons, plus two more RDM-DMX outputs. Both connect to the host computer via USB, directly if the USB output is butch enough, via a powered USB hub or, with the VX20, by connecting mains power via an IEC cable if not. [VX20 Wing] [VX20 Trackball] [VX20 Faders]

Metal made, heavy enough to feel serious and to not slide around a desk when in use, yet easy to carry with a useful finger-hold under the front edge, the does VX20 resemble a mini lighting console. The trackball, LED backlit and changing colour according to function, has space around it to rest the hand (for right-handed users, at least) while only being a short reach to the parameter buttons, and though there is no armrest the front edge is nicely curved to wrap your hand around. The trackball can be switched between operating as a mouse (albeit one that feels a little 'slow') or as a parameter encoder, and there are then keys for stepping through or clearing selections, changing parameter types (curiously color here while colour in the software!), switching to blind, and recording cues.

Still missing, though, is the ability to deal with more than one parameter at a time, since you still effectively only have one encoder - the trackball. Plus you still sometimes have to reach for the computer, notably for a shift key for range selections, and for a numeric keypad: Clarity tries to downplay channel numbers, but they are sometimes useful, plus you often need to specify cue numbers. How long before someone offers a bracket to let a keyboard float at the back of the VX20? They're not dissimilar in width. Or until a multi-encoder VX30 wing appears?

Integration with the software is nicely done: a powered wing glows its trackball red until it finds its software, while each DMX output has a green LED that only lights when that output is actually enabled in the software.

### In Action

As with others reviewed recently, this is a console at an early stage of its life - so there is a mixture of high-power features useful to some and which look good at trade shows (pixel grid, media server integration), features that will be brilliant in the right context (the performance window, which those who busk shows live will adore), clever touches (draggable time editing, the clever range displays), and then the things so obvious they have somehow just been forgotten (record cue-only). Plus it crashed a few times in our hands, though it never lost show data.

Of course, the ratio of missing to brilliant will evolve over time. What was remarkable, though - particularly given the complexity of control on offer - was how easy it was to get going, even when meeting Clarity for the first time. And, beyond that, to get a long way into very complex sequences before even realising the complexity you'd achieved. That is quite an achievement; true, it's partly brought about by applying the now-familiar computer interface and conventions to a lighting control, but there's no harm in that. Clarity is a powerful lighting tool.

Its software basis means you will also need to consider the console-vs-computer factor. LSC are offering some keen pricing at the moment, and computers are cheap. A laptop-plus-wing makes for a powerful, portable lighting combination, but that laptop might be a bigger theft target than a lighting console, for example (like most laptops, the wings include Kensington lock points allowing you to tie everything down), and you'll be the one making sure the wing is happy talking to your computer, rather than the manufacturer of an integrated console.

That said, if your happy to deal with that (and in most cases, it should just work, since the hardware and software are designed as a pair) - or if you're actually after a lighting control you can run on your laptop - you'll be very happy with the facilities offered by Clarity, especially relative to its price.

Plus, of course, you can download it yourself and try it out before spending any money at all . . .

[www.lscighting.com/clarity-trial](http://www.lscighting.com/clarity-trial)

**NB. Since this article was written, LSC Lighting has informed L&S of a number of enhancements that will be available in the next release, including Tracking Backup, a Channel Controller for TV lighting operation, the addition of the Record Cue Only function, enhancements for the trackball control and the ability to map extra user interface buttons to the wings.**