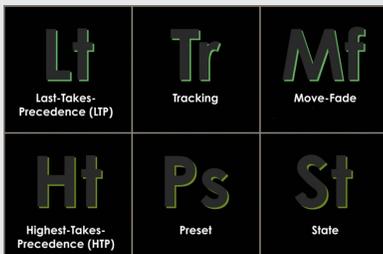




Entertainment Lighting Control Philosophy

by Anne Valentino and Sarah Clausen, ETC Control Product Managers



Introduction

Is there a new lighting control console in your future? Will you be buying – or selling – or using – such a console? How do you know which one is right for you – your team – or your customer? Beyond channel counts, DMX outputs, faders, touch screens and buttons described in most console datasheets and product comparison articles, how can you find out if a lighting control console will “feel right”? If you ask friends and colleagues, they may say “buy console x because it is easy to use” or “buy console y because it thinks like you do”. Why is one console perceived to be easier to use than another or to “think like I do”?

We hope to clarify some of this by providing you with tools to truly evaluate consoles based on their core philosophies rather than things like channel counts and fader quantities. Let’s get to the bottom of why there are so many different consoles out there and why many of them enjoy a devoted following. When considering operational philosophies, there is no right or wrong; there is only different.

The philosophy of a console defines its basic personality. All of the basic philosophies in use on lighting desks today are derived from our original task of controlling intensity. These ideas have been modified to extend to moving light and media control, but they all have strong historical precedence. It is worthwhile to understand where these ideas come from, as this understanding can help you make a choice about the product that might be right for you.

There are many styles of production out there, and what may be a disadvantage to one style may be a great advantage to another. We make no claims that one style is universally better than another and in no way mean to offend the reader if what we classify as a disadvantage is seen as a clear advantage by you. Having said that, let’s begin...

What is a Core Philosophy?

The core philosophy of a desk can be described by answering three basic questions:

- 1) What gets stored in a record action and how is that data edited?
- 2) How are cues played back during the performance?
- 3) How do multiple control sources (inputs) interact with one another?



Computerized lighting control systems began to appear in the mid-to-late 70s. At this point, most of the production world was using one of two different types of manual analog control – either resistance dimmers (piano boards) or multi-scene preset desks. When the first computerized desks were developed, they used the ideas already in place from these systems. Effectively, they simply computerized the way people were already cueing their lighting on manual desks.

The operation and cueing styles piano boards and multi-scene preset systems were substantially different, and gave rise to the same differences in desks that we still see today. Words like tracking, preset, last take precedence (LTP), highest takes precedence (HTP), move fade and state were all indicators of the functional differences between piano boards and multi-scene preset desks. Over time, those words have become confused. People will often talk about tracking, LTP and move fade as though they are the same thing. They are not. There is a perception that placing a tracking console in cue-only mode makes it into a preset style desk. This is not the case.

The explanations below will hopefully help bring clarity to this.

What gets stored?

There are two essential schools of thought here – TRACKING and PRESET. Each of these methods gets their start at the very beginning of manual lighting control.

Tracking Philosophy

Tracking comes from the use of resistance dimmers– a very “hands on” approach where each dimmer had a physical handle for control. Moving the handle up and down would increase or reduce the resistance on the electrical circuit, causing the attached lamps to dim. Control was actually at the dimmer rack, which was typically very large and very hot. Electricians crews – and it took multiple electricians to run one of these setups – would move the handles in real time to create cue transitions.

If you looked at the cue sheets for a show run on a piano board, each cue contained only information about the handles that had to be moved to create the cue. If a level needed to be edited, it would be changed in the cue that contained the move. The new level would then “track” through subsequent cues until another move instruction was written down – this was the ultimate linear “single scene” controller.

CORE IDEA for TRACKING: Once a channel is at a value, it remains at that value until you give it a new move instruction - therefore any modifications to the level are automatically propagated forward until a new move instruction is encountered.

Preset Philosophy

Preset style came from the introduction of remote control for dimmers. Each dimmer still had an individual handle, however this handle was a small low voltage potentiometer (or “pot” or simply “fader”) mounted on a control panel. The actual dimmers were located in another room, far away from the stage, so they could be noisy and hot someplace else. With this remote control came the ability to set up multiple scenes and crossfade from one to the other. In a





two-scene preset system each dimmer has two handles, one in each scene – a five-scene preset desk has five handles for each dimmer, and so forth. When writing down the cues for this type of system, all handles that needed to be set above zero must be included, whether they are changing level or not. This is because the input from one scene is faded out while the input from the next scene is faded in. If you don't set the fader for the unchanging channel, it will fade out completely – not the intent of the designer .

CORE IDEA for PRESET: Each cue in a preset system is a snapshot of the stage look. When editing, any changes to a dimmer value through a series of cues has to be edited in each individual cue.

Why do these differences matter?

This determination of whether a console is a tracking console or preset console used to be the main distinction among theatrical “conventional” consoles. Everyone was controlling the same thing – intensity – and this difference between tracking and preset really defined how a designer could work.

A tracking system made it very easy to create fade-within-fade transitions. Think of one electrician moving his handles in a 5 count while another electrician moves his handles in a 10 count. Tracking consoles have offered this kind of control from the beginning since their goal was to recreate this manual control style in a computerized way.

Preset consoles have a harder time performing this kind of task since the basis of their behavior is to cross-fade from one whole scene to another, usually with only two scene masters. The fade for each scene is controlled by one handle, so getting multiple fade times is impossible.

This simple issue leads us to the next basic question.

How are cues played back?

This question can open up a very complicated discussion of multiple cue lists, masking, priority settings, etc., but it all comes down to two essential choices: MOVE FADE or STATE.

Tracking consoles were historically also Move Fade desks. Because a cue contains only the “move instructions”, when the cue is executed only those move instructions are replayed. If a level is in progress on a long fade from cue 10, and that channel tracks through cue 11, it will continue to fade in cue 10's time when you press go, even if cue 11 has totally different fade timing. This is because the tracked level in cue 11 does not take over the move instruction of cue 10.

CORE IDEA for MOVE FADE: When a cue is executed, only move instructions are played back and tracking levels (those values unchanged from the previous cue) do not affect a change. Tracking levels that are still in a timed transition from a previous cue will keep moving using the timing from the previous cue.

You can see how Move Fade and Tracking are closely related, but not quite the same. So, how does this idea play out in Preset desks?





The alternative style of playback is to execute the entire contents of the cue on a go instruction. We refer to this as a STATE machine. Preset consoles are typically state machines since they really don't pay any attention to tracked levels. So, if you take the example above and switch to a preset way of thinking, the timing of cue 11 will "stomp" on the fade of that channel from cue 10 because in essence all cues contain move instructions for all channels.

CORE IDEA for STATE: When a cue is executed, all channels are played back every time. Within each cue execution, all channels stored in the cue receive new timing data, even if they are at the same level as the previous cue.

Again, you can see how Preset and State are closely related. One idea (Preset) is how the data gets stored, and the second (State) refers to how it is played back.

How do multiple control sources interact with each other?

What outputs impact the current stage state are drawn from the "Move Fade" and "State" concepts described above. However, it is rarely that simple.

Almost every computerized lighting console has more than one source of control for any channel. At the very least, there is a manual control source and a cue list. Add in a single submaster, and you now have three potential control sources.

How these control sources interact with each other is very important from a playback point of view. The two basic methods – last take precedence (LTP) and highest takes precedence (HTP) – have their origin in those same resistance dimmer boards and preset consoles of the past.

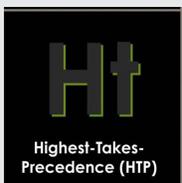
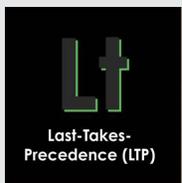
In attempting to describe the logic of a fade engine with multiple control inputs, we often use words like "LTP winner" and "HTP winner."

The resistance dimmer board is the ultimate single-scene controller. This system is about as LTP (Last Takes Precedence) or "Last Action" as it gets. Because each dimmer has only one control handle, it makes sense that the output of the dimmer will always be the last setting you made. This basic idea was maintained when computerized desks were developed that DID have multiple control sources.

CORE IDEA for LTP: Channels always do the last thing you told them to do, regardless of which control source told them to do it.

Preset consoles, with their columns of faders all controlling the same dimmer, required a different control method. The idea chosen was HTP (Highest Takes Precedence). The fader with the highest overall output level, based on the setting of the fader and the master for that fader's scene, would "win". If that fader was taken down, or that scene was faded to a lower level, the next highest fader/scene master combo would "win" and the channel would sit at that level until either the fader/scene was faded down or another HTP "winner" came around.

CORE IDEA for HTP: Channels determine their output level by calculating which control source has the highest level. The highest level wins.





HTP and LTP are typically only an issue for intensity control. Even desks today that default to HTP intensity use LTP for non-intensity parameters, since HTP is meaningless in regard to pan, tilt, color, etc. Since a console that uses HTP as its intensity philosophy can affect playback in significant ways, it remains a defining characteristic.

Almost all desks today support a blend of LTP/HTP options for intensity control. As such, the concept of the LTP/HTP winner becomes very important in answering the question that is asked ALL OF TIME. "Why is that light at that level?" Typically, desks calculate the LTP "winner" first, look to the "HTP" winner to see if the level is higher or lower and then output the resultant level. Well, almost, because that level could be further refined by inhibitive submasters and then once again by grand master values.

A final word (or two) on "philosophy"...

Prior to the advent of moving light desks, a console was either a TRACKING-MOVE FADE-LTP or a PRESET-STATE-HTP machine. These philosophies went hand-in-hand and remained linked together. Hopefully, from the descriptions above, it is easier to understand why.

The first moving light desks broke the above standard associations. You may find a moving light desk that is Tracking/State by default, or that is Preset/Move Fade. And that is because the control of moving lights is inherently more complex than the control of just single parameter devices (dimmers).

An additional clarification may be helpful - Tracking versus track editing. Almost all desks – even preset desks – can support track editing. This is the idea that you can make a change in a cue, and force that change forward through the cue list until a different value is encountered. This is track editing. It is a function that can be specifically called for in a preset desk when editing. It is the way tracking desks work by default. All tracking desks support "Cue Only" functions, which make edits in the specified "cue only," thereby not allowing the edit to "track forward." In this way, a tracking desk can be made to edit data in a manner similar to preset systems. HOWEVER, placing a tracking console in "Cue Only" mode does not turn the console into a preset desk. It simply affects how edited data will be handled.

Tracking and preset desks differ in how they get their content in the first place – and this basic behavior is not changed just because a tracking desk is in Cue Only mode. For example, assume you have written cues 1 thru 10. Those cues contain channels 1 thru 10. You add channel 100 at full to the stage. You then do a selective store to create cue 4.5. Depending on your desk, this will be along the lines of [100 Record 4.5].

On a preset desk, in cue 4.5, only channel 100 will be "on". Channels 1 thru 10 will be driven to zero. On a tracking desk, channel 100 will be "on", and channels 1 thru 10 will track in at the level they have in cue 4.

So, what happens in cue 5? In a preset console, channel 100 will be at zero. In a tracking desk, if the desk is in track mode, channel 100 will track in at full. If you recorded the cue and appended the "cue only" button, or if the desk was in "Cue Only" mode, channel 100 will be driven to zero.

Lt Last-Takes-Precedence (LTP)	Tr Tracking	Mf Move-Fade
Ht Highest-Takes-Precedence (HTP)	Ps Preset	St State



Finally, because almost all modern desks have more than one Go button, new concerns about storing the data arise. If you are using multiple playback faders and multiple cue lists, it is no longer appropriate in most instances to store the entire state of the rig in a cue. Storing the entire state of the system essentially means you can have only one cue on stage at a time – which defeats the concept of multiple cue lists. So, different mechanisms have been developed to manage this rather sticky issue.

What gets stored - revisited.

To facilitate the use of multiple cue lists, two early concepts were developed for moving light desks to further refine the answer to the question “what data am I storing in the cue?”

One idea was that only lights that were active (In use) should be stored in the cue. You’ll see this idea played out as an option in a number of different desks today. By default, the entire state of the light will be stored. Commonly, desks using this approach are preset desks.

The second idea was “the programmer”. When parameters are modified live, they are pulled into the Programmer. When a cue is stored, only the values in the Programmer are stored as move instructions in the cue. Desks with programmers are always also tracking desks. As such, they follow the rules outlined above.

Selective Storage methods

Filtering and Masking

The complexity of lighting today means that the even more control is needed when storing and recalling record targets (cues, submasters, palettes [see below]). As such, most desks support the concept of filtering or masking data. This means simply that the console gives you a way to be selective about what you record or play back. Some consoles allow this at the category level – focus, color and beam – while others allow you to filter or mask down to the parameter level – pan, cyan, gobo, iris, etc.

Filtering and masking essentially do the same thing – limit what gets recalled or recorded. The means by which they do this, however, is essentially opposite. When you apply a filter, you are actively selecting the data you are interested in keeping. When you apply a mask, you are actively selecting the data you wish to ignore.

Filters and masks are very useful for repetitive storing activity. For example, if you are building a series of cues that you only want to store color data in, you probably have the lights on. But you don’t want to store their intensity. So, you can filter color or mask everything but color to store just the color data.

Command Instructions

Most desks also supply syntax that can restrict what is stored. This syntax will vary, depending on the desk, but it is along the lines of the following (using the same example as above):

- [Color] [Record] [Target] or
- [-] [Intensity] [Record] [Target]



Some other ideas worth mentioning

Referenced data

Another feature that moving light desks gave us has to do with the oft-times repetitive nature of working with automated lighting. It quickly became apparent that the act of setting lights manually to the down stage left sofa each time you wanted them there, or putting them in the closest approximation of R80 over and over again was time consuming and, well... boring. Enter referenced data, typically called palettes. These smaller record targets are building blocks upon which cues can be built. Palettes are constructed from “absolute data” – the term that we use for lights that have values provided from encoders and/or a keypad (channel 5 at 50, cyan at 35, magenta at 20 and yellow at 0 are examples of absolute data).

Palettes are typically built for intensity, focus, color and beam parameters separately. This automatic filtering by category makes it faster to set them up. Most desks support “all palettes”, which allow you to put any data into one of these referenced building blocks. Typically, you focus any lights you might want on the down stage left sofa on that sofa and store them in focus palette n. Then, when you later need one of your lights on the sofa, you just select it and place it in focus palette n. When the director later moves the sofa three feet to the left at the last dress rehearsal, you are left with one focus palette to update instead of 300 individual cues. The modification to this focus palette then propagates through all of the cues in which it was used.

Avoiding that light that swings down from the rafters, changing colors and gobos on the way...

It is very common when using moving lights to preset the lights in their required pan/tilt, color and beam values prior to the light fading its intensity up. This is often called “marking” the light.

The act of marking lights to specific locations in the cue list is a major, often very time-consuming job for the moving light programmer. All desks now support some type of marking function, to varying degrees of usefulness.

Most desks now also provide automatic methods to preset the lights with no user intervention. You may hear terms such as “Move While Dark”, “Move Before Bright”, “Go in B”, “Dark Move” and “Auto-Mark” for this activity. This is typically a desk setting that can be enabled or disabled, generally with a cue level override to allow live moves (lights fading up while non-intensity parameters move).

Desks also often support more controlled styles of marking, where the programmer makes more definitive decisions about where the light will preset (or mark).



How to deal with the light that does everything...

Finally, moving lights also require complex definitions – called by various names throughout the industry (profiles, libraries, templates, personalities, etc). These fixture definitions contain the DMX mapping of all the parameters and can also include further details about those parameters, like frame positions, macro functions, and other bits of data. Some consoles use a normalization that allows you to enter and record data in real world units, like a Hz value for strobe, degree measurements for pan and tilt or hue and saturation data for color. This makes swapping fixtures from different manufacturers very easy as the console takes on the burden of translating those real world values into the actual DMX levels required to achieve the desired look.

Real World Examples

So, let's put these theoretical ideas into perspective. How do some ETC consoles fit into these classifications?

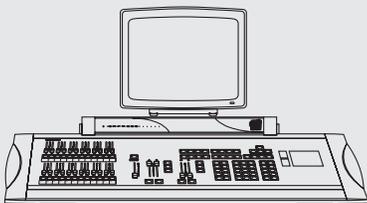
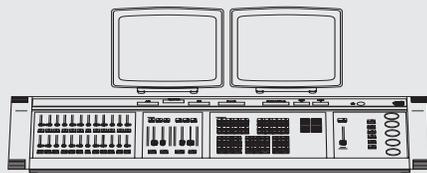
ETC Expression

Expression (this classification includes all Expression-style consoles including Insight and Express) is a PRESET-STATE-HTP conventional lighting console at its core. It has one cue list and two playback fader pairs for control of that cue list. In addition, Expression offers a number of submasters for playback of specific types of data – a single look or a chase effect, for example.

Expression offers a flexible command structure for data entry and its philosophy allows for simple out-of-order playback of cues. The HTP nature of the intensity control makes it easy to layer looks from different submaster faders and it's relatively easy to see what will happen on stage based on the current state of those faders.

When using the keypad to enter channel levels, Expression will “capture” those channels in order to have the manual changes override the HTP nature of the console. Captured levels stay captured until released by the user. When recording a cue, Expression takes a snapshot of the entire output of the console unless other commands or settings are used (independent submasters or “Except” commands). In addition, Expression does not auto-play the new cue when it is recorded. If you play back cue 1, then make manual changes and record cues 2 through 10, when you release those captured channels you will find yourself back in cue 1.

When editing cues in Expression, users typically capture the channels they wish to adjust and then make a series of Update or Record commands to get those changes into a series of cues. Expression offers “track editing” functions to make mass changes across cues in the cue list. Because Expression is not a TRACKING console, however, few users feel truly comfortable using this feature.



Expression 3 added some moving light support, like fixture personalities, Focus Points, LTP behavior for non-intensity parameters and background fading of LTP channels for fade-within-fade operation. While these additions made it easier to control moving lights from an Expression-style console, it did not change the core philosophy of the console. Also, Expression 3 continues to use individual control channels for each parameter on a moving light – a “horizontal distribution” causing a 20-parameter fixture to use up 20 control channels on the console.



Where Expression becomes less useful is if you need fade-within-fade control. While something similar can be accomplished with part cues (channels can be assigned to cue “parts” and each part can have its own timing) you still cannot press GO on the next cue until all parts have completed without stomping on the timing of those parts. To get a ‘true’ fade-within-fade running on an Expression, you can use another playback (via a linked submaster) or you can plan carefully to allow the long fade to happen in the second crossfader pair while you run subsequent cues in the main crossfader pair.

ETC Obsession

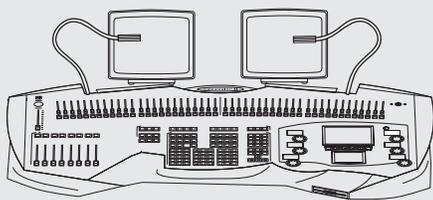
Obsession is a TRACKING – MOVE FADE – LTP conventional lighting console at its core. It has one cue list with multiple playback faders to accommodate fade-within-fade operation. Each new cue gets executed in a new fader with its own timing intact. Because the console differentiates between move instructions and tracked values, each new move runs in its own timing without affecting other long-running fades. Obsession has 8 playback faders on the face panel and 118 virtual faders to allow for many fade-within-fade operations to simultaneously occur on stage.

Using the keypad to enter levels does not capture any of those values; they are simply manual. The interaction between the keypad and faders follows move fade/LTP rules. A manual value will stay manual until a cue is executed that provides that channel a new move instruction. Again, it is important here to notice the difference between a move instruction and a tracked value.

Obsession’s command syntax creates a sentence that is similar to the way a lighting designer speaks. This is a strict command line that expects a certain order of commands every time. While a user’s first experiences with this style of syntax may be somewhat trying (since the console will stop your data entry with a “syntax error” when nonsensical commands are entered) it does make it very clear what kind of commands the console expects to see. Once learned, this syntax becomes very clear and can be used to make intricate changes in one command line.

Obsession’s tracking style and single cue list make it an excellent choice for productions with a clear beginning, middle and end – in other words, a linear production like theatre. The fact that it has a single cue list - allowing only one resultant look on stage at a time makes it less suited for more random playback scenarios. It is worth noting that all tracking systems long ago overcame problems running cues out of sequence. There is a misconception that running a cue out of sequence on this style of desk would not give you the same results as fading to that cue in sequence. This is not the case at all.

Obsession in this case also includes the Obsession II, which added some features to support the control of moving lights, such as encoders for manual control, Focus Groups, and displays that made it easier to see the settings of non-intensity parameters. However those parameters were still distributed horizontally (a 20-channel device took up 20 control channels) and the core functionality of the console was not changed. In addition, Obsession offers very little control over what gets recorded into a cue or Focus Group beyond whole fixtures.





ETC Congo

The ETC Congo is a PRESET – STATE – HTP lighting control console with TRACKING tendencies. Let us explain...

The ETC Congo follows the Avab philosophy, which mixes these core ideas in interesting and challenging ways. The design idea behind Congo came from understanding how European theatres work and the desire to add moving light control to a theatrical console. European theatres traditionally work in rep with rep plots and very often rehearse out of order. Because of this, Congo is a PRESET-STATE-HTP console for intensity. This cannot be changed. However, Congo treats non-intensity parameters in an LTP manner.

Congo has different option settings that change how non-intensity parameters can be recorded. The default is to record all parameters of channels that are “active” – have intensity above 0% - and to not record anything for channels that are out. Because Congo has a non-traditional cue list structure, it lends itself to non-linear playback and reuse of Presets (the recorded stage look) which usually works better if those Presets are recorded as snapshots (in PRESET style). Because a total stage state is most often NOT what you want to record, however, the use of intensity as a recording flag makes it very clear and predictable what will be recorded at any given time. Congo can also be set to record only changed (moved) parameters in a tracking style.

Congo offers a main playback for any sequence (cue list) with full playback controls (Go, Go Back, Pause, Rate controls, Seq + and Seq -, crossfaders) and additional Master Playbacks with more limited playback controls. Master Playbacks are pageable in banks of 20 and can contain almost any content in the play (show file).

Congo’s show structure is very database-like with cues built from building blocks - palettes for moving light parameters, presets for what has traditionally been known as cue data. A cue in Congo is an event, not a target, based on how a preset is used. When a preset is assigned directly to a master fader, it behaves like a submaster (with the addition of moving light data). When a preset is assigned to a sequence (cue list) step and played back using a Go button, you get a cue. Congo supports multiple sequences and takes the idea of referenced data to new heights. In essence, everything is referenced data in Congo. Presets can reference palettes, Sequences reference presets. Playback controls reference their content – no piece of Congo data is inherently owned by any playback control (except for Independent faders and switches).

Congo offers an extremely flexible short command direct-access structure for data entry and its philosophy is ideal for out-of-order playback of cues. Within a sequence, Congo will always play back the resultant state for out-of-order cues. While you can achieve fade-within-fade for moving light parameters, it is more difficult to achieve this with intensity, and jumping out of order between playbacks will play back the complete intensity state. How you recorded these presets in the first place can make Congo’s behavior appear to be state or move fade for moving light parameters.

When using the keypad to enter channel levels, Congo will not “capture” those channels by default. Intensity entries at the keypad will be in the HTP pool unless the Capture mode is enabled. Captured levels stay captured until released by the user. Congo does auto-play the new preset when it is recorded.





Congo's short-command syntax and non-traditional cue structure make it very fast to program and provide very flexible playback options. Once Congo's basic command syntax and structure are learned, the console becomes very intuitive to use. It is necessary for the person programming a Congo to interpret the designer's commands since not many designers will speak in Congo syntax. Also, how one plans to play back and edit the show data should inform which method one uses to record that data in the first place.

ETC Eos

The ETC Eos is a TRACKING-MOVE-LTP console, with HTP options for intensity only. At first glance, Eos seems decidedly Obsession-like. This is largely driven by the fact that Eos (and the other members of the Eos family) is philosophically the same as Obsession. In the core areas of operation, the syntax is very similar.

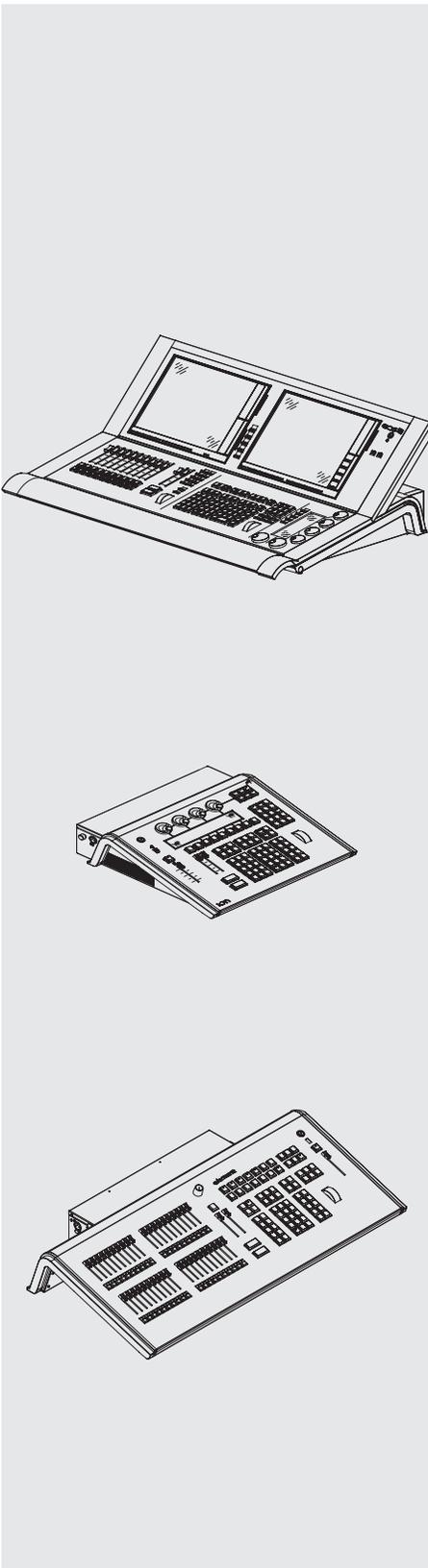
The basic syntax of command line systems was easily extendable to address moving lights. This was fortuitous for a number of reasons. First, over the years, the syntax of the Obsession and Expression product lines has become increasingly similar. So we knew we had a platform that users of traditional ETC products would be comfortable with. Also, then, Eos and Congo have vastly different operational styles (in the same way the Obsession is a style different from Expression). Not believing that one size fits all, having two modern, powerful product lines like Eos and Congo in the ETC product portfolio gives users a choice as to which philosophy and style suits them best, in the same way they could make a choice between Obsession and Expression.

Because there always need to be exceptions to the rule, Eos offers a few things to modify the default behavior of the desk. First, you can place the desk in Tracking or Cue Only mode. It's worth noting that placing the desk in Cue Only mode does not turn Eos into a preset console, it simply means that any modifications to cues or cues recorded out of sequence within an existing cue list will not impact subsequent cues. This is an editing function. The basic tracking behavior for playback is maintained.

Eos offers Independent status for faders and submasters. This means that a parameter owned by an independent fader cannot be stolen by a non-independent fader. Independent status is shared equally; so two or more faders on independent can trade off ownership.

In general, when multiple cue lists are active, the move fade philosophy makes it very clear what gets played back. Only moves are executed, no matter what fader generates those moves. Tracked values are not played back, so no fader priorities are required. However, Eos provides an Assert function for forcing tracked levels to play back when desired. Asserts can be placed at a cue level, a cue part level, a channel or a channel parameter level.

Eos is, by default, a fully LTP system in regards to multiple cue lists and the interaction with the keypad. Because it is sometimes useful to be able to hold onto a manually set level, it is possible to "capture" manual values, withholding them from playback activity. Non-intensity parameters are always handled in LTP fashion. By default, cue lists also manage intensity LTP, although it is possible to set an HTP flag for intensity values in a cue list. Submasters, by default, handle intensity HTP, but can be changed to work LTP.





Earlier, we discussed how desks make the decision about what should be recorded into cues. Eos (very deliberately) does not have a Programmer, but it does have two basic record commands – Record and Record Only. The Record function stores all parameters for any channels that are not at their default position. Those could be manual values, values from submasters or values from cue lists. The Record command assumes you are interested in the entire state of any luminaires that have been moved from home.

Record Only, on the other hand, behaves like a Programmer, recording only manually set values and ignoring (filtering out) untouched parameters and parameters that are getting their current value from the playbacks. These two methods allow for a solution to the linear playback problem of ensuring that what you see on stage is what you will get when you play back the cue, but also allows for selective storing of specific data for overlapping or more random playback.

Further ways to decide what you are going to store in a record target include using filters: set a filter for “Beam” and only the beam data will be stored. Within Beam, you could set a filter on just “Iris” and only the iris data will be stored. The same logic for Record and Record Only applies. If you set an iris filter and use Record, all iris values that are not at their home position will be stored. If you use Record Only then only the manual iris values will be stored. Selective storing via the command line – where you specify the parameters that you are interested in (or are not interested in), provide a further method of modifying “what is stored?”

Like almost all tracking consoles, when you take a cue out of sequence, either through a Go to Cue instruction or a link, the entire contents of the cue are replayed.

To conclude...

There are many, many consoles on the market today, all of which control intensity and moving lights, all of which can create effects, all of which can play back cues.

Think about how the desk will be used – will it be used for theatre or events? Linear or random playback? By professional or novice users? What consoles have those people used in the past? What types of production will they be doing in the future? Is it a rep house? Does the venue do a lot of fast turn one-nighters? All of these questions come into play when selecting a lighting desk.

Our goal is to give you two completely different, completely capable control systems with products spanning the market from budget-minded small venues to the largest opera houses and performing arts centers, with each of those families representing a different mind-set.

~~~~~

If you have any questions or comments, we welcome them. You can contact us at [anne.valentino@etcconnect.com](mailto:anne.valentino@etcconnect.com) and [sarah.clausen@etcconnect.com](mailto:sarah.clausen@etcconnect.com).