E&S Lighting

Documentation · 1.5.1-Alpha

Installation

E&S Lighting is free lighting control software, inspired by grandMA in the way it is used. It drives your fixtures over **Art-Net** and **sACN** from a computer, with a mouse, a keyboard or your finger on a touchscreen. No hardware is required.

The software is in **Alpha**: it is usable, but features are still changing and bugs are possible. The **Report a bug / an improvement** button, at the top of the window, lets you write to us in two clicks.

Recommended configuration

	Minimum	Recommended
System	Windows 10 64-bit	Windows 11 64-bit
Processor	2 cores	4 cores or more
Memory	4 GB	8 GB
Screen	1024 × 600	1920 × 1080, touch
Network	an Ethernet card for the DMX output	a card dedicated to the lighting network

Linux and macOS versions are in preparation: you can request them from the download page of your account (**Request** button); the most requested come first.

The lighting calculation runs on every processor core except two: the more cores, the bigger the shows the software can handle (thousands of fixtures and universes).

Getting the software

1. Create an account on the license website: this is your license request. Say in a few words how you will use it (theatre, concerts, events…).
2. Wait for approval: it is done by hand, you will see it on the **My account** page.
3. Once the license is active, the **My account** page offers the installer (Setup) for your system. Download it.

Installing on Windows

1. Run `ESLighting-Setup…exe`. Windows may show a "Windows protected your PC" warning: the software is not signed (a signature costs several hundred euros a year). Click **More info**, then **Run anyway**.
2. Follow the installer. An **E&S Lighting** shortcut is created in the Start menu and on the desktop.
3. On first start, Windows may ask to allow the software on the network (firewall): allow it for **private** networks, otherwise no DMX frame will go out.

Where your files are

What	Where
Your shows (`.esl`)	`Documents\E&S Lighting\Shows`
Exported profiles (`.elsf`)	`Documents\E&S Lighting\Profils`

Computer settings, fixture library, crash reports	`%AppData%\ESLighting`

A show is a single file: to back it up or move it to another computer, copy its `.esl`.

Updating

When a new version comes out, the software tells you at startup (**New version** popup). Download the new Setup from **My account** and install it over the old one: your shows and settings are kept.

If a version becomes too old to be used, the software says so and the DMX output is cut as for a deactivated license: install the latest version.

Uninstalling

Control Panel → Programs → **E&S Lighting** → Uninstall. Your shows in `Documents` are not deleted.

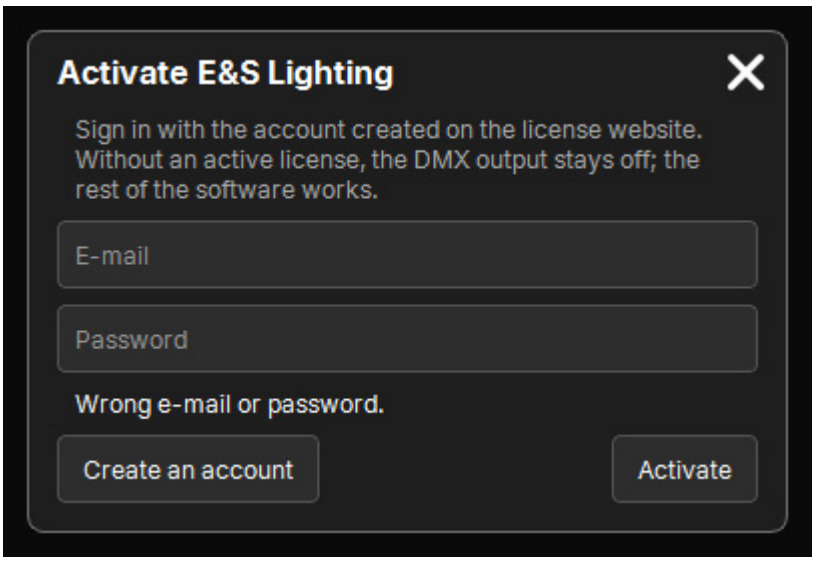
Next: activate the license.

License

The license is free. It tells us who uses the software during the Alpha and lets us keep you informed. Without an active license, **the whole software works** (programming, cues, layouts…) except the **DMX output**, which stays off.

Activating

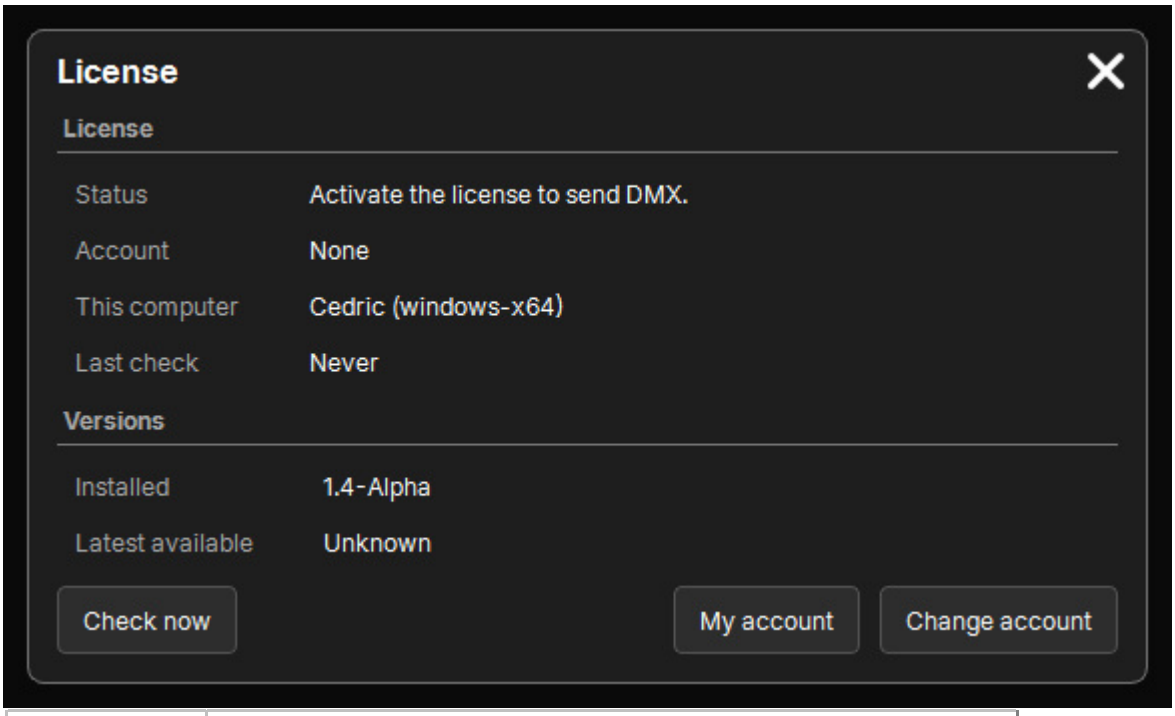
On first start, the **Activate E&S Lighting** popup opens. Enter the e-mail and password of your account on the license website, then **Activate**.



- No account yet: **Create an account** opens the website.
- The computer is registered on your license: **2 computers** at most per license.
- The password is not kept: the computer receives a key of its own (a token). Changing your password or e-mail on the website therefore does not cut computers already activated.

The License popup

"Add a window" → **Setup** → **License** shows the state of the license:



Line	What it says
Status	Active, or why the output is cut
Account	the account e-mail
This computer	the computer name and its system
Last check	the last answer from the server
Offline	how long the software can still run without a check
Versions	the installed version, the latest available, the minimum required

- **Check now** asks the server right away (after a reactivation on the website, for example).
- **My account** opens your page on the website.
- **Change account** opens the activation again.

Offline

The software checks the license at startup, then every 24 hours. Without internet, it keeps working normally **for 30 days** after the last successful check; it warns you from day 25. A connection, even a short one, is enough to start again for 30 days.

Attention: setting the computer clock back to gain time blocks the output ("Computer clock behind") until the clock is set right again.

Before a cut-off

If the license must be cut (deactivated on the website, computer removed, version too old), the software never cuts in the middle of a show without warning:

- a popup **30 minutes** before, then **5 minutes** before;
- at 5 minutes, **Postpone 1 h** gives one more hour, once;

- once cut, the DMX output sends nothing, and a gray veil over the Layouts reminds you (**Activate the license** button).

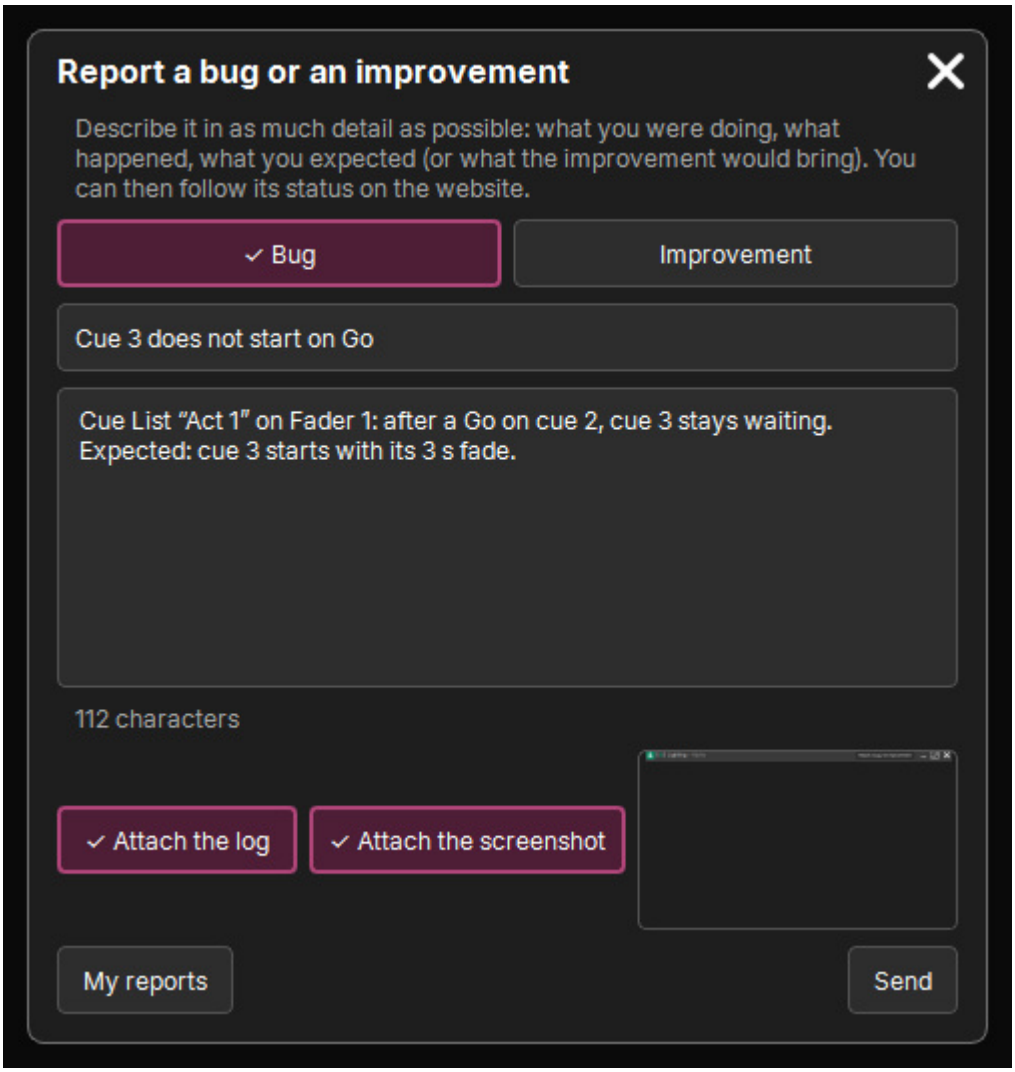
Changing computers

On the website, **My account** page → **Activated computers**:

- **Remove** frees the place: the old computer will cut its output at its next check (after the 30-minute delay).
- Then activate the software on the new computer.
- A removed computer stays in **Removed computers**: **Reactivate** (if there is a place left) or **Delete**.

Reporting a bug or suggesting an improvement

The **Report a bug / an improvement** button, at the top of the window, sends your message to your account on the website:



1. Choose **Bug** or **Improvement**, give a short title.
2. Describe it in as much detail as possible: what you were doing, what happened, what you expected.

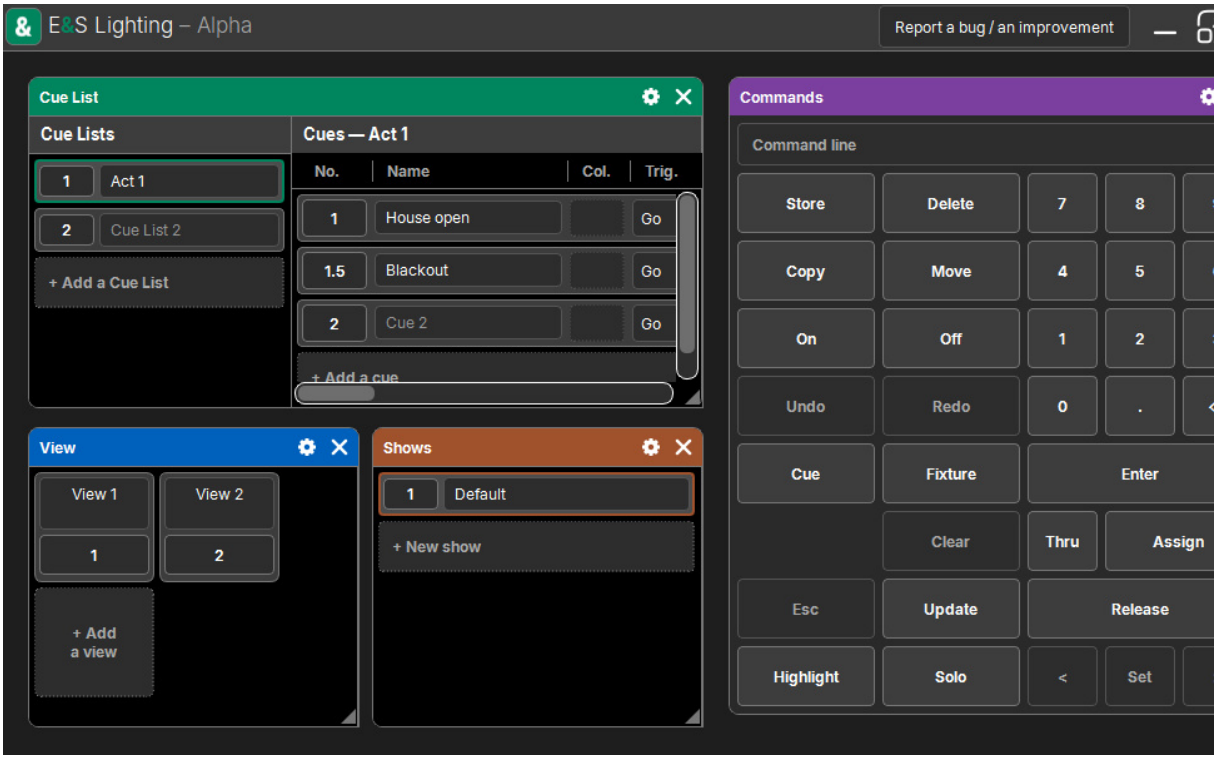
3. Leave **Attach the log** (the last lines of the software log and the crash reports of the week) and **Attach the screenshot** (the screen at the moment of the click) checked: they help a lot.
4. **Send**. Without internet, the message is kept and goes out on its own at the next license check.

You then follow its status on the website, **My reports** page: accepted, in progress, done (with the version that brings it) or declined, with the reason.

After an unexpected shutdown of the software, the **Unexpected shutdown** popup at the next start offers **Report**: the crash report goes out with your description.

The screen and windows

E&S Lighting has no menus: the screen is a **free area** where you place the windows you need (Cue List, Programmer, Layout, command pad…), at the size you want. An arrangement is kept in a **View** so you can recall it in one move.



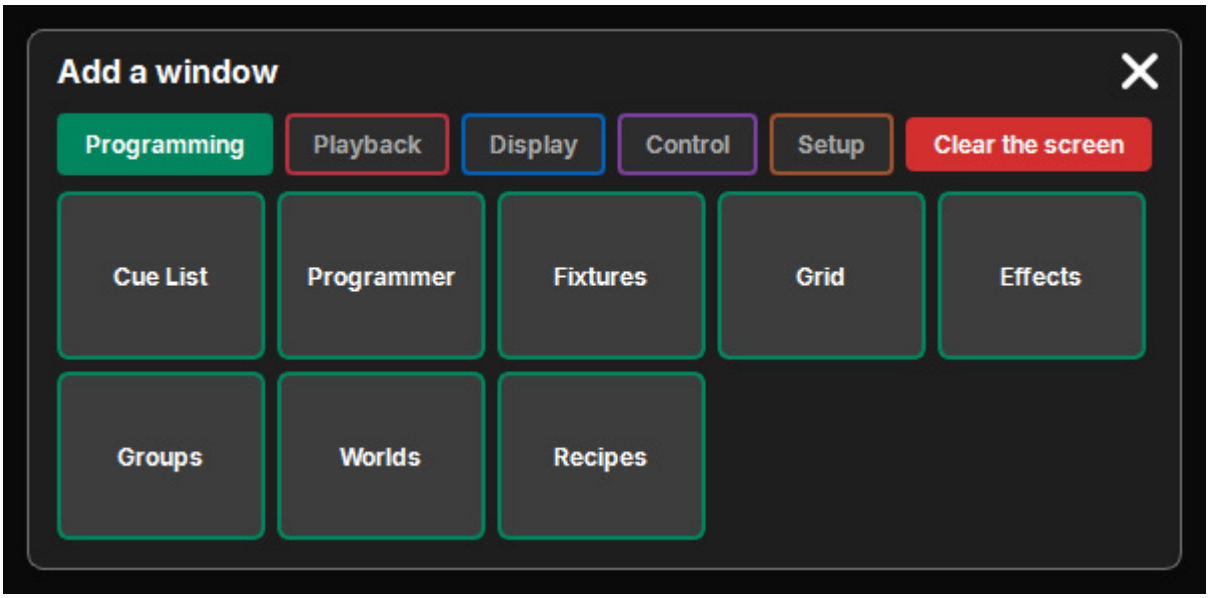
The header

At the top of the main window:

- the logo and the name, followed by the version stage ("– Alpha");
- on a secondary screen, the name of that screen;
- **Report a bug / an improvement** (see License);
- minimize, maximize, close. **F11** switches full screen on and off.

Adding a window

Tap (or click) an **empty spot** of the area: the **Add a window** popup opens. The chosen window will open where you tapped.



Windows are sorted into categories, each with its color, which you find again on the window headers:

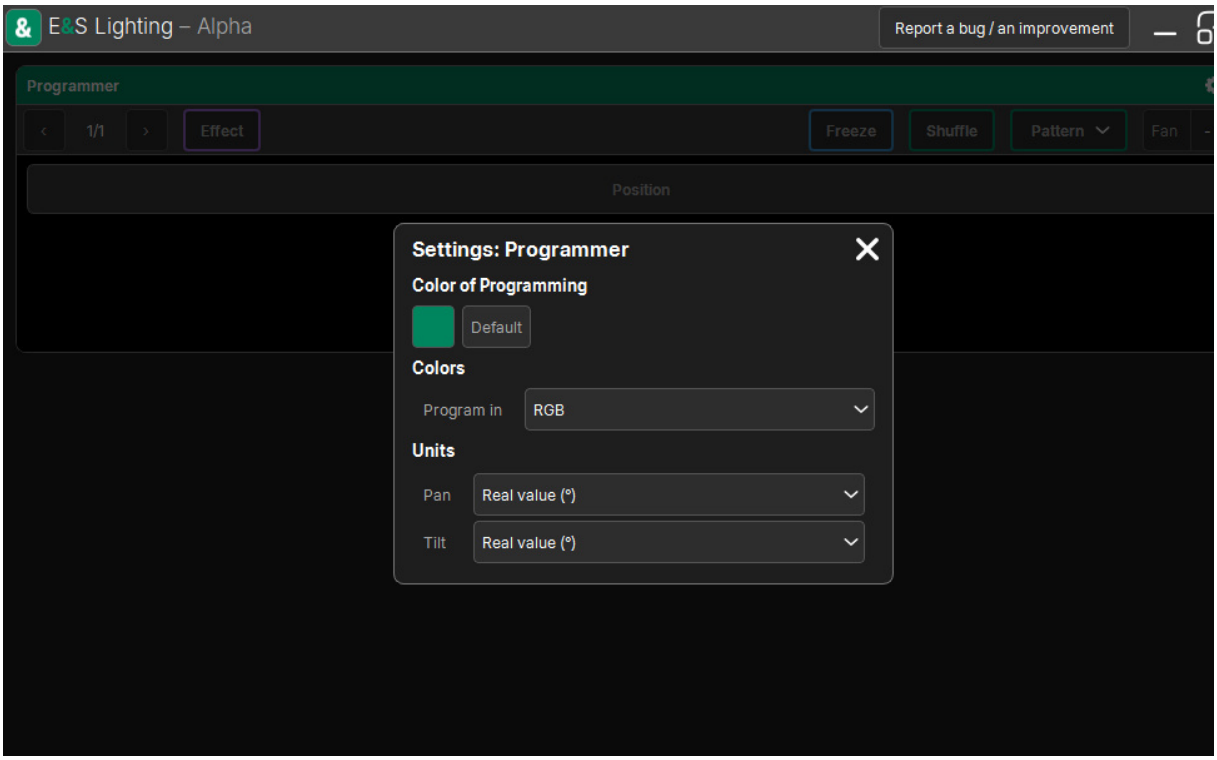
Category	Windows
Programming	Cue List, Programmer, Fixtures, Grid, Effects, Groups, Worlds, Recipes
Presets	one preset pool per attribute family of the patch (Dimmer, Position, Color...)
Playback	Faders, Buttons, Speed, Timecode
Display	Layout, Image, View, Windows, Clock
Control	Commands, Actions, History
Setup	Shows, Attributes, Profiles, Patch, DMX Output, DMX Input, Time Input, About, License, Language

The last category chosen is remembered. **Clear the screen** (in red) closes every window of this screen.

Moving and resizing

- **Move**: drag the window header.
- **Resize**: drag the small triangle in the bottom right corner. The window edges only serve its content, you never grab them by mistake.
- A window **never** covers another one: when you move or enlarge it, it stops on contact with its neighbor and slides along it.
- The cross in the header closes the window.

The gear



The gear in the header opens the window settings:

- the **color of its category** (a preference of the computer, applied to every window of the category);
- then what is specific to the window: columns shown, units, colors in RGB or CMY, etc.

These settings are kept with the window, and in Views.

Several screens

The **Windows** window (Display) manages the main windows: one per connected screen, each with its area and its windows. A screen is set **On** or **Off**. Views recall all screens at once, or only those they include.

Lists and tables

Most windows show lists of cards (cues, groups, profiles...):

- a card has a **name** you type directly, and a **number button** used as the object for the command pad (see the principles);
- **+ Add...** at the bottom of the list creates the next object, without a popup;
- on a touchscreen, **one finger** draws a selection rectangle, **two fingers** scroll.

Language

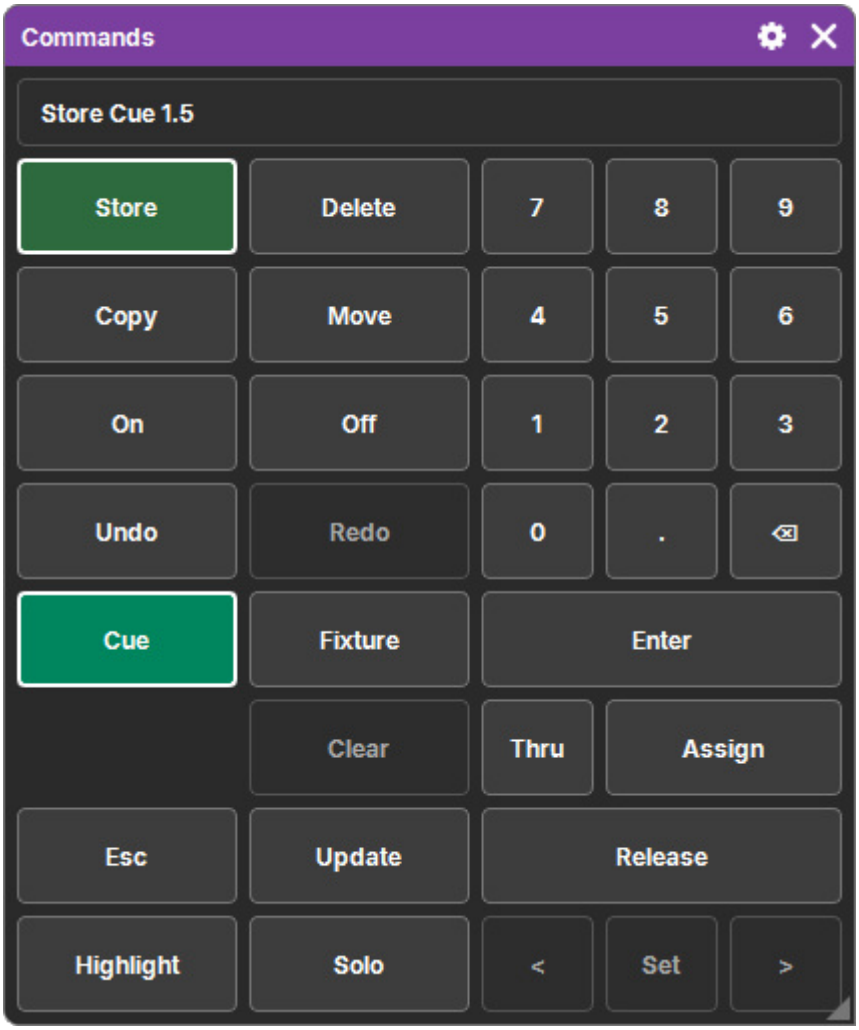
"Add a window" → Setup → **Language**: Français or English. The change applies at the next start. By default, the software uses the Windows language.

The command pad

The command pad is the on-screen equivalent of a console surface. Open it with "Add a window" → **Control** → **Commands**. It always keeps the same size and its keys never move: you find them by touch.

E&S Lighting follows the grandMA rule: **action first, object second**. Press **Delete**, then the cue to delete; **Store**, then the group to store into. With no action chosen, pressing an object does its normal job (select the cue, recall the group, load the view...).

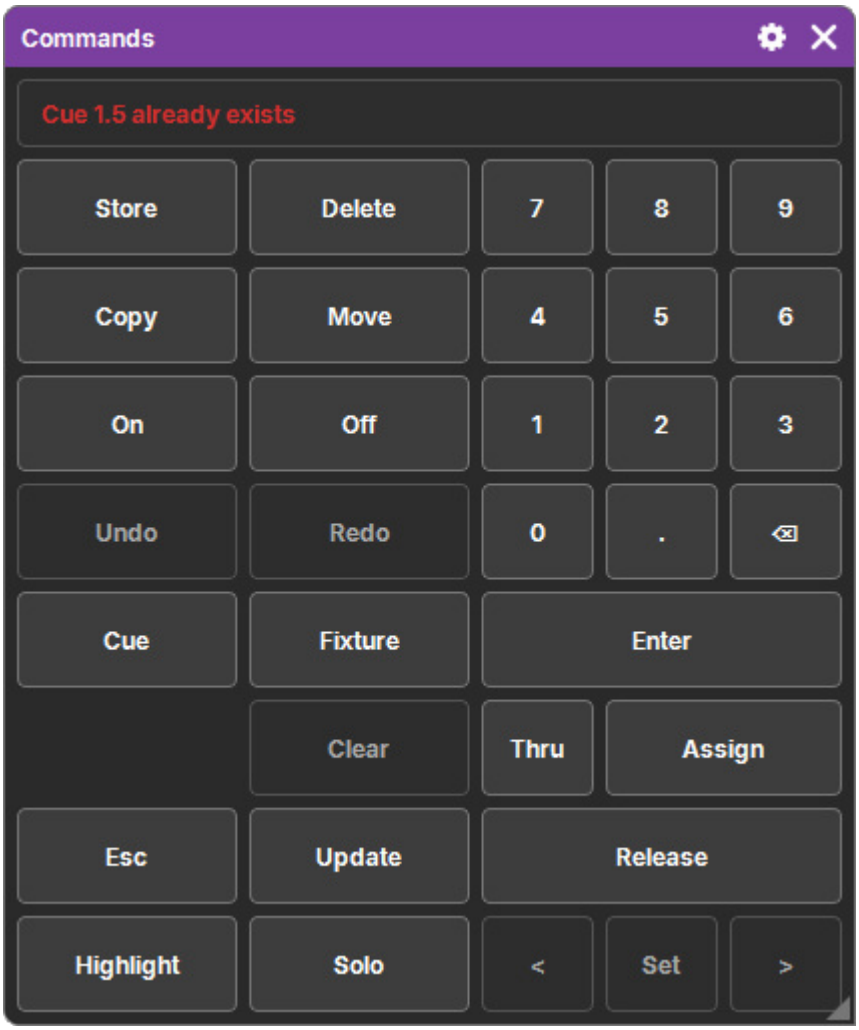
The command line



At the top of the pad, the command line shows what is in progress: "Store Cue 1.5", "Copy 3 views at...", "Fixture 1 Thru 50".

- Grayed out ("Command line"): nothing is in progress.
- In white: the command you are composing.
- In red: the command was refused, and why ("Cue 7 does not exist", "No Cue List shown"...). The message stays until the next key.

Most messages of the software go through this line: keep a pad open somewhere.



Action keys

An action key that is lit waits for its object; it takes its color, and the buttons of the objects it can act on take the same color. Pressing the lit key again turns it off.

Key	What it does
Store	Stores the programmer into the object pressed next: a cue, a group, a preset, a view, a Fader or a Button (in its Cue List). On an object that already exists, a question offers **Replace**, **Merge**...
Delete	Deletes the pressed object, or the whole selection if the object is part of it. With a selection, **Enter** is enough.
Copy	First press on an object: the source. Second press: the target, where the copy arrives. **Enter** alone: after the last object.
Move	Like Copy, but the source leaves its place.
On	Turns on: a main window that is off, a Fader's Cue List (immediately, without a fade), an attribute or a family in the programmer (at its current value).
Off	Turns off: a Cue List, a main window; in the programmer, removes

	fixtures or attributes.
Assign	Assigns one object to another: a Cue List to a Fader or a Button, an object in a Layout, a group in a World.
Release	The targeted attributes get a **Release** in the programmer: stored in a cue, it hands control back to what lies underneath.

The details of Delete, Copy and Move, common to every list, are in Lists, tables and objects.

Park and Unpark

A **quick double press** on **Off** turns it into **Park**; on **On**, into **Unpark**. Park freezes fixtures at what they are outputting right now: nothing goes over them anymore (neither Cue List nor programmer), only the grand master still applies. Unpark frees them.

- **Park Fixture 1 Thru 5** then **Enter**: these fixtures.
- **Park** then **Enter** alone: the programmer selection.
- A double press on **Fixture** writes "Park Selection": **Enter** parks the selection.

See Park.

Objects: Cue and Fixture

Cue

Cue, a number, **Enter** acts on the **Cue List shown** (the one chosen in the Cue List window):

Command	Effect
Cue 5 Enter	selects cue 5
Store Cue 5 Enter	stores the programmer into cue 5 (created if it does not exist, otherwise Replace / Merge)
Store Cue 2 Thru 8 Enter	stores into every existing cue from 2 to 8, in one go
Delete Cue 5 Enter	deletes cue 5
Copy / **Move**, press a cue, **Cue 7 Enter**	the cue (or the selected cues) arrives at 7

A cue number has up to 4 digits and 2 decimals: **1**, **1.5**, **12.25**. The keyboard comma counts as a point.

Fixture

Fixture, a number, **Enter** selects fixtures in the programmer:

Input	Fixtures
Fixture 5	fixture 5
Fixture 1 Thru 50	from 1 to 50, in this order
Fixture 50 Thru 1	the same, from 50 to 1 (the order matters for spreads)
Fixture 101.3	sub-fixture 3 of fixture 101 (an LED bar, a multi-cell

	fixture)
Fixture 101.1 Thru 104.8	the sub-fixtures, from 101.1 to 104.8
Fixture 101.0	the head of fixture 101 alone (without its sub-fixtures)

What does not exist in the patch is skipped. With an action:

- **Off Fixture 1 Thru 5 Enter**: they leave the programmer and the selection;
- **Release Fixture 1 Thru 5 Enter**: all their attributes get a Release;
- **Off Enter** alone, **Release Enter** alone: the programmer selection.

The numeric keypad

Key	What it does
0 to **9**, **.**	the object number, or a value
Thru	a range: "1 Thru 50"
⌫	erases the last character; otherwise the object; otherwise the action
Enter	runs the command line

A refused digit (an impossible number: too many digits, two points) is simply not written.

Typing into a field

When a field is being edited (a name, a Programmer value, an address...), the digits, **.**, ⌫ and **Enter** of the pad go **into that field**, like a keyboard: you can set everything with your finger, without a physical keyboard. Pressing the pad does not lose the editing in progress.

A value for the selection

With selected cells that accept a value (for example the Patch addresses), digits typed without an action are written in the command line with the name of what they set ("Address 1.1"). **Enter** gives the value to the whole selection, **in the order the cells were selected**: for addresses, the first gets 1.1 and the next ones follow. See Selecting.

Undo and Redo

Undo undoes the last change to the show; if a command is in progress on the pad, it cancels it first. **Redo** redoes what was just undone. They are grayed out when there is nothing to do. See Undo and Redo.

Clear and Esc

Two keys to "erase", which do not target the same thing.

Clear: the programmer

Clear only touches the programmer, in three steps, as on grandMA:

1. if there are selected fixtures (or a Pattern): the selection is released;
2. otherwise, if there are **active** values: they are deactivated. They stay in the programmer (the light does not move), but will no longer go into a store;
3. otherwise: the programmer is emptied, effects included.

If you are going through the selection fixture by fixture (**< >**), Clear first goes back to the whole selection. Grayed out: there is nothing to clear.

Esc: everything else

Esc (or the keyboard Esc key) acts **outside the programmer**, in two steps:

1. if there are selected cells (in a list, a table, or objects of a Layout in Set mode): it only deselects them;
2. otherwise: it cancels what is in progress on the pad (the lit action, the command line, the red message).

Highlight, Solo, Set

Key	What it does
Highlight	toggle: the selection goes to its **Highlight** values (defined per channel in the profile: full intensity, open white...), above everything. To find fixtures.
Solo	toggle: the intensity of all **other** fixtures goes to 0. Without a selection: nothing.
< >	go through the selection fixture by fixture: what you set only applies to the current fixture. In a loop.
Set	one press: back to the whole selection; another: back to the sub-selection you came from.

Holding Set with **>** or **<** splits the selection: 1 in 2, 1 in 3... (grandMA's Interleave); **<** and **>** then go from one part to the next. With a mouse or a single finger you cannot hold Set and press elsewhere: **hold Set for half a second** then release it, it stays lit until the next press. These three keys are grayed out without a selection.

Update

Update opens a popup with what the programmer can update: first the **presets** its active values come from (even if adjusted since), then the **current cue** of each Cue List that is playing. Each card says what it is ("Color 3", "Cue 2"), on which executors it plays and which attributes change; **Update** applies it. Only what the target already contains is updated: nothing is added. With nothing to update, the popup says so.

Selecting

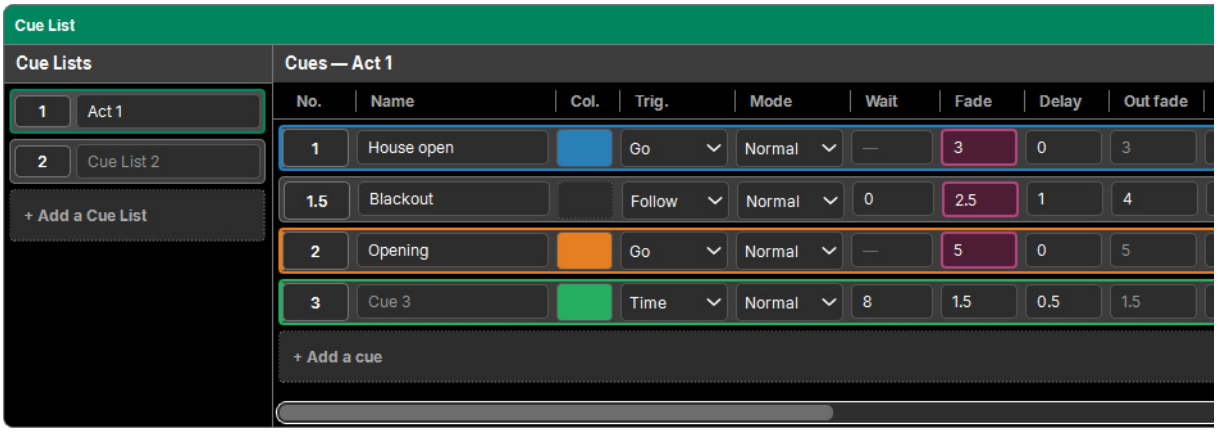
There are two kinds of selection in E&S Lighting:

- the **cell selection**: cues, rows, table cells, pool tiles... to change or copy them all at once;
- the programmer **fixture selection**: the fixtures you are setting.

The cell selection

There is **no Select mode** to turn on: you select directly, with the mouse or a finger.

Selecting



Drag from a cell (or from the background of the list): a dotted rectangle is drawn, and every cell it touches is selected. They take a selection color.

- The selection keeps **the order**: the starting cell first, then the others in the order the rectangle touched them (top to bottom, left to right for those touched at the same time). This order matters for a value typed in sequence (see below).
- Near the edge of the list, it scrolls on its own and the rectangle follows.
- A selection holds only one kind of cell: cues or groups, not both. Cells of another kind than the first are refused.
- For a single cell: a small drag on it.
- In some tables (cues, profile channels, Patch, DMX Output, DMX Input, Timecode...), pressing a **column title** selects the whole column, from top to bottom.

A new rectangle replaces the previous selection.

Editing the whole selection

- **Pressing a selected cell** edits it for **the whole selection**: choosing a value in a drop-down list, a color, a mode... applies to every selected cell, in one go (a single Undo).
- A selected text field does not open for editing: type the value on the pad or the keyboard, it is written in the command line, then **Enter**.
- **Pressing elsewhere** (on a cell that is not selected) empties the selection, then that cell alone is edited normally.

Exception: a pad action waiting for its target (Copy, Move...) keeps the selection, since it uses it.

A value in sequence

Select cells that take a value, type it on the pad without an action, then **Enter**: each cell receives it, in selection order. What follows on goes in sequence:

- Patch addresses "1.1": 1.1, then the next free address for each fixture;
- IP addresses "10.0.0.11": .11, .12, .13...;
- universes, channel numbers: in sequence.

Other values are the same for all. The command line shows what will be set ("Address 1.1") before **Enter**.

Touch

- **One finger** sliding draws the selection rectangle.
- **Two fingers** scroll the list, in both directions.
- A short tap of one finger is a normal press.

A single finger waits a very short moment before it counts (in case the second one arrives to scroll): this is what avoids selecting or opening a field when you meant to scroll.

Mouse

- Drag with the left button: the rectangle.
- Wheel: scrolls. A drop-down list does not change value with the wheel while it is closed, so nothing changes when scrolling a table.
- A click on a text field does not open it for editing right away if you start dragging: you can draw a rectangle starting from a name.

Emptying the selection

Esc (or the Esc key): a first press empties the selection; the next one cancels the action in progress on the pad. See The command pad.

The fixture selection

This is the **programmer** selection: the fixtures whose attributes you set (see The Programmer). You make it:

- **on the pad**: **Fixture 1 Thru 50 Enter**, or a number, a sub-fixture (**101.3**), a range in the order you want (**50 Thru 1**). See The command pad;
- in a **Layout**: pressing a fixture selects it (another press deselects it); a rectangle drawn on the background (mouse in **Select** mode) adds the fixtures it touches, in drawing order. See Layout;
- with **Groups**: pressing a group selects its fixtures, in the group order. See Groups.

The **Fixtures** window shows the selection (highlighted rows, **Select** filter) but does not make one.

The selection adds up... then starts over

As on grandMA: **as long as no value has been changed**, a new selection adds to the previous one (a group, then another, then fixtures on the pad). **As soon as a value has changed**, the next selection starts from scratch; the values set stay in the

programmer.

Removing and stepping through

- **Clear** releases the selection (first step), then deactivates and empties the programmer (see Clear).
- **Off** then a group, a fixture of a Layout, or **Fixture N Enter**: these fixtures leave the selection and the programmer.
- **< >** step through the selection fixture by fixture, **Set** goes back to the whole selection. See Highlight, Solo, Set.

Lists, tables and objects

Almost everything in E&S Lighting is a **numbered object**: a cue, a Cue List, a group, a preset, a view, an Action... Windows show them in two ways, with the same gestures everywhere.

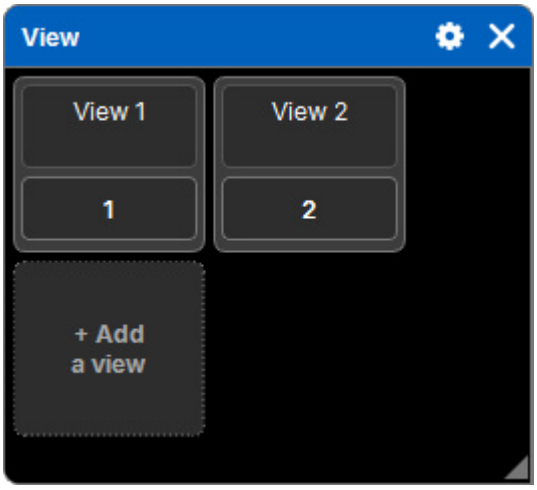
Cards

A list (Cue List, Shows, Windows, Profiles...) is made of **cards**, sorted by number. A card has:

- a **number button**: this is the object, for the command pad. Without an action, it does the normal job of the object (open the show, select the cue...); with an action lit on the pad, it receives it (**Delete**, **Copy**...);
- a **name**, typed directly in the card. It is saved as soon as you confirm (**Enter** or by leaving the field).

Tables (cues, profile channels, Patch, DMX Output...) are cards with several **columns**, with their titles above.

Tiles



Pools (Views, Groups, Presets, Actions, Images...) show their objects as **tiles**, in a grid: a square with the number, and the name underneath, on two lines, typed directly. The grid arranges itself according to the window width.

Adding

The last card (or the last tile) is **+ Add...**: it creates the next object, after the last one, **without a popup**. You then name it in its card. An addition never fills a gap in the numbering: it always goes after the last one.

Columns

- **Width**: drag the handle on the right of a column title. Widths are kept with the window (and in Views).
- **The whole column**: in several tables, pressing the title selects the whole column (see Selecting).
- **Sorting**: in the **Fixtures** window, pressing the title of **No.**, **Address** or **Profile** sorts the rows: ascending, then descending, then the original order. An arrow

shows the direction.

A table too wide for its window scrolls horizontally; the titles follow.

Delete, Copy, Move

The same sequence applies to **every** list and every pool.

Delete

- **Delete** then an object: it is deleted. If the object is part of the selection, **the whole selection** is.
- Or: select first, **Delete**, then **Enter**.

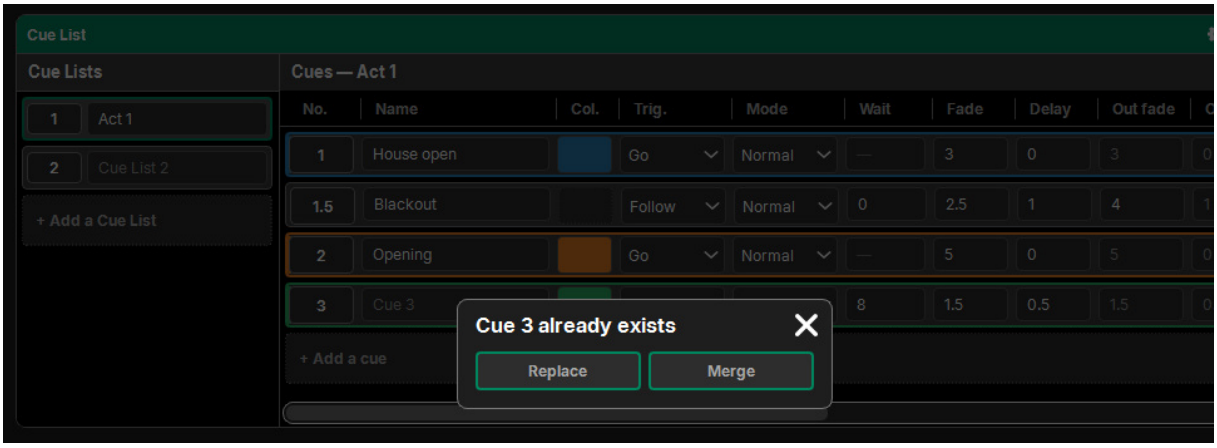
Copy and Move

1. **Copy** (or **Move**), then an object: this is the **source**. If it is part of the selection, the whole selection is the source. The command line writes it: "Copy 3 views".
2. Then press the **target**: the source arrives there.

Or: select first, then **Copy**: the selection immediately becomes the source.

- **Enter** instead of a target: the source arrives **after the last object** of the list.
- Copied objects keep their gaps: cues 1, 1.1 and 2 copied to 5 arrive at 5, 5.1 and 6.
- The target must be in the same list as the source.
- For cues, you can also type the target: **Copy**, a cue, **Cue 7 Enter**.
- Profiles have no number: **Copy** then a profile (or **Enter**) makes a copy of it, "(copy)".

Replace, Merge, Append



If objects already exist where the source arrives, a question opens:

Choice	Effect
Replace	the source replaces the objects that were there
Merge	the source is added to what was there (cues only: their values merge)
Append	the source arrives right after the target; if those numbers are taken, all the following objects are shifted to make room

The cross of the question cancels. Arriving on a free number asks nothing.

The whole operation is **a single** history entry: one **Undo** undoes all of it.

Drop-down list



Choices (a mode, a unit, a network interface…) are drop-down lists:

- the list opens **below** the cell (or above, if there is no room) and stays inside its window; when longer, it scrolls by dragging;
- the **wheel** only changes the value if the list is open: scrolling a table never changes anything by accident;
- on a selected cell, the choice applies to the whole selection.

Popups

A question or a setting opens as a **popup** over the window it comes from: only that window is veiled, the pad and the other windows stay usable (for example to type a value). A popup too big for its window takes the whole screen. The **cross** closes the popup; for a question, it **cancels**.

Undo and Redo

Every change to the show can be undone: deleting a cue, renaming a group, changing an address, storing a preset... E&S Lighting keeps the **history** of what you have done since the show was opened.

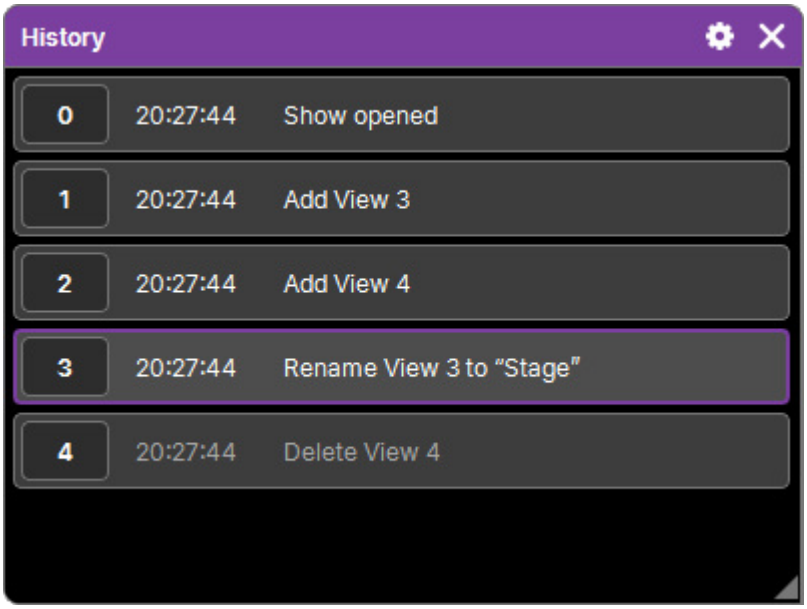
Undoing, redoing

How	Effect
Undo on the pad, or **Ctrl+Z**	undoes the last change
Redo on the pad, or **Ctrl+Y**	redoes the last undone change

- If a command is in progress on the pad (a lit action, a command line started), **Undo** cancels it first, without touching the show.
- The keys are grayed out when there is nothing to undo or redo.
- In a field being edited, Ctrl+Z and Ctrl+Y belong to the field (they undo what you are typing).
- Making a new change after undoing clears what could still be redone.

What you see as **a single action** is a single entry: deleting 12 selected cues, copying 3 views, giving a value to a whole column... is undone with a single **Undo**.

The History window



"Add a window" → **Control** → **History** shows the list of changes, from the oldest (**0: Show opened**) to the most recent. Each row: its number, its time, what it did ("Delete Cue 3", "Rename Cue List 2 to "Act 1""...).

- The current position is highlighted; what was undone (and can be redone) is dimmed.
- **Pressing a row goes back to that moment**: everything in between is undone, or redone, at once.

What goes into the history

Everything that changes the **show** and that you meant to do: objects created, deleted, copied, moved, renamed; stores (cues, groups, presets, views); patch, show profiles, settings of Cue Lists, Faders, DMX outputs; main windows (On / Off, names); Park and Unpark…

What does not

- the **selection** (cells, fixtures);
- the **programmer** (values, effects being set): it is a workspace; **Clear** handles it;
- what **plays**: Go, Fader, Flash, Off of a Cue List, Actions run, triggers by Timecode or DMX Input (but if an Action **stores** or **names** an object, that does go in);
- the place and size of open windows (kept separately, see Windows and Views);
- computer preferences: language, category colors, license.

Limits

The history belongs to the **session**:

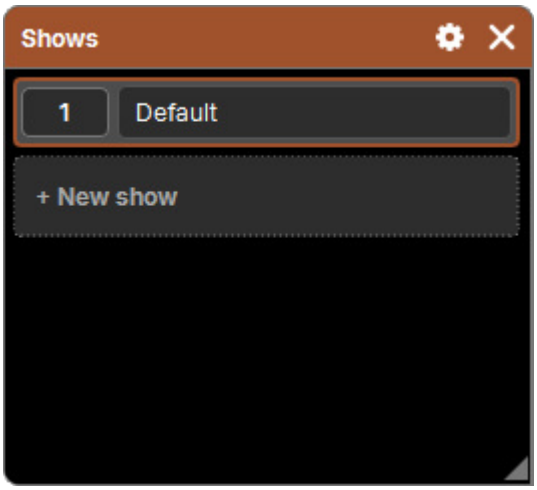
- it is emptied when you open another show;
- it keeps the last **200** changes (fewer if they take a lot of memory, for example cues with thousands of values); the oldest ones go away on their own.

The show itself is saved as you go: there is nothing to "save" before closing. See Shows.

Shows

A **show** contains everything you prepare for a performance: the patch, the cues, the presets, the layouts, the window arrangement... A show is **a single** `.esl` **file**. Only one show is open at a time.

The Shows window



"Add a window" → **Setup** → **Shows** lists the shows of the folder, with the open show highlighted.

Gesture	**Effect**
press a show's button	opens it (the open show is closed)
+ New show	creates an empty show, "Show 1", "Show 2"... (it is not opened)
type in its name	renames it (the file too)
Copy then a show	makes a copy of it, "My show (copy)"; the open show can be copied
Delete then a show	moves it to the **trash**; refused for the open show

The open show can be renamed too. A refused name goes back to the old one: empty, already taken, more than 64 characters, a space at the start or the end, or a character not allowed in a file name (`\ / : * ? " < > |`).

Saving

There is **no Save button**: every change is written to the show file the moment you make it. You can close the software at any time without losing anything.

If a write fails (disk full, USB stick removed, file no longer accessible), the **Show not saved** popup warns you: your last changes may be lost on closing. Make a copy of the show as soon as possible.

At startup

The software reopens the **last open show**, with its windows as you left them. If it has disappeared, it opens the first one in the list; if there is none, it creates the **Default** show.

If the remembered show cannot be opened (damaged file, locked, read-only, or created by a newer version of the software), the **Show not opened** popup says so, and another show is opened instead: the software never starts without a show.

If some cues could not be read back (damaged file), the **Damaged cues** popup lists them: what is missing stays empty, the rest of the show is untouched. Storing one of these cues again replaces what was damaged.

Files

What	Where
the shows	`Documents\E&S Lighting\Shows\<name>.esl`
the trash	`Documents\E&S Lighting\Shows\Corbeille` (the show name followed by the date and time)

- **Back up**: copy the `.esl` file (with the software closed, or with **Copy** in the Shows window for a safe copy while it is open).
- **Move to another computer**: copy the `.esl` into the Shows folder of the other computer, it appears in the Shows window.
- **Recover a deleted show**: take it out of the Corbeille (trash) folder and rename it `<name>.esl`.

What the show keeps, what the computer keeps

In the show (`.esl`)	On the computer
the patch, and a copy of the profiles it uses	the **library**: families, attributes, fixture profiles
the Cue Lists and their cues, the Faders and Buttons	the language, the category colors
groups, presets, effects, Worlds, Speed Masters, Park	the license
layouts, images, views, Actions, Timecodes	the GDTF Share account (its password, in the system keychain)
DMX outputs, DMX and Time inputs	the display of DMX values, the last open show
the main windows, their place, and the open windows	

A show therefore carries a copy of the profiles of its patch: it opens as is on another computer, even if its library is different. See Profiles.

Not in the show: the programmer (it starts empty on opening) and the history (see Undo and Redo).

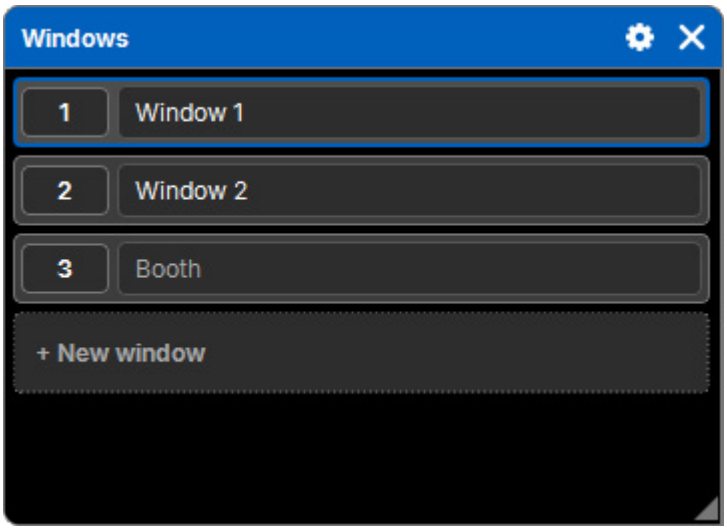
Windows and Views

Two levels of windows:

- the **main windows**: one per screen ("Window 1", "Window 2"...), each with its header and its area;
- the **windows** you place in an area: Cue List, Programmer, Layout... (see The screen and windows).

A **View** keeps the window arrangement so you can recall it in one move.

Several screens: the Windows window



"Add a window" → **Display** → **Windows** lists the main windows of the show. The one holding the list is highlighted; windows that are off have their name dimmed.

Gesture	Effect
+ New window	adds a main window, switched on, "Window N"; place it on the screen you want
press a window's button	brings it to the front (and switches it on if it was off)
On / **Off** then a window	switches it on / off
Delete then a window	deletes it, with its windows
type in its name	renames it; the name shows in its header

- **Window 1** is always there and always on: it cannot be switched off or deleted. Closing it (cross in the header) **quits the software**.
- Closing another main window with its cross only **switches it off**.
- A window that is off is hidden but keeps its windows: switched back on, it finds them again.
- Each main window remembers its place: the screen (monitor), the position, the size, maximized or full screen. It goes back there when the show opens. A new window (or a window whose screen was unplugged) goes by default to the monitor with the same number if it exists, centered.

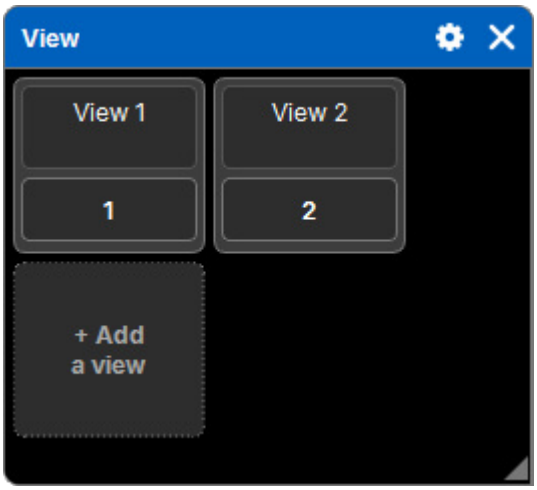
Main windows are part of the show: switching on, off, renaming or deleting goes

through the history.

The current arrangement

Open windows, their place, their size and their settings (column widths, filters, Clock color...) are kept in the show at all times: on reopening, everything comes back as you left it. This is not a View, and it does not go into the history.

Views

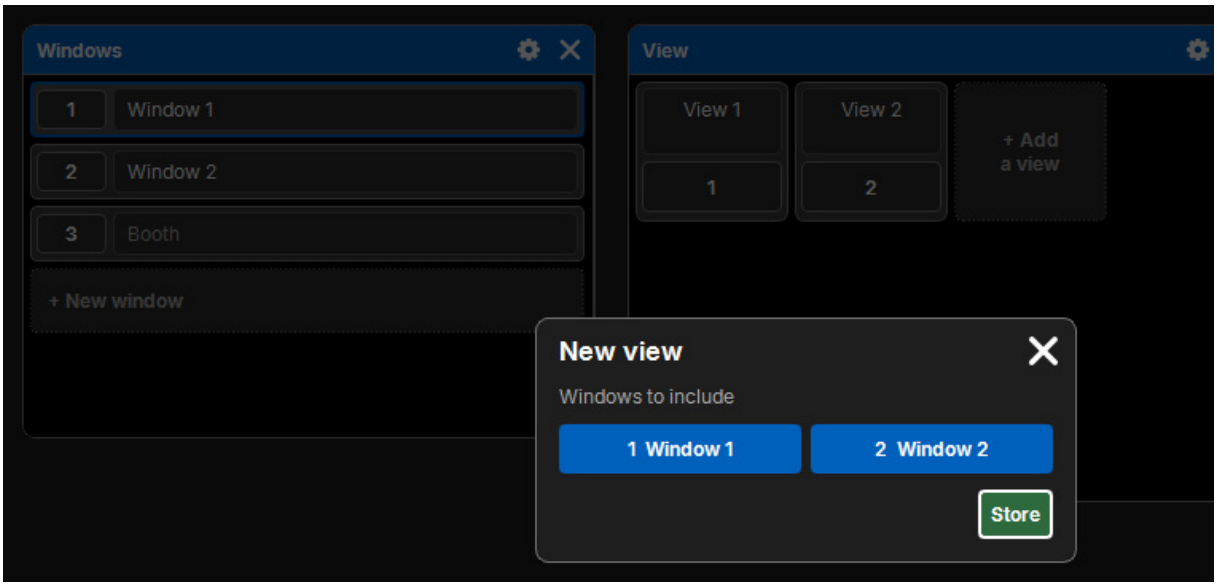


"Add a window" → **Display** → **View** shows the views as tiles, sorted by number.

Gesture	**Effect**
press a view	**recalls** it: the windows of its screens are replaced by its own
+ Add a view	stores the current arrangement into a new view, at the end
Store then a view	stores the current arrangement into it; if it exists, a question offers **Replace** or **Append** (a new view right after)
Delete, **Copy**, **Move**	the common sequence (see Lists, tables and objects)
type in its name	renames it

A view keeps, for each window: its type, its place, its size and its settings.

A view's screens



A view keeps **the screens it includes** and, when recalled, **replaces only those**: the other screens do not move. You can thus have views for the programming screen, and others for the playback screen.

- With a single main window on, a view includes that screen.
- With several main windows on, storing opens **Windows to include**: one key per window that is on, all pressed by default; release those not to include, then **Store**. The cross cancels.
- A window that is off is never included.

Views are part of the show: storing, renaming or deleting them goes through the history. A view can also be placed in a Layout, to recall it from a lighting plot.

Attributes

An **attribute** is something a fixture can do: Dimmer, Pan, Tilt, Red, Gobo 1, Zoom... Attributes are grouped into **families** (Dimmer, Position, Color, Gobo, Beam...), like the Feature Groups of grandMA and GDTF. The channels of profiles point to an attribute, and it is through attributes that the Programmer, presets and effects talk to fixtures, whatever their brand.

The **Attributes** window ("Add a window" → **Setup**) is the only place where families and attributes are created: profiles only pick them.

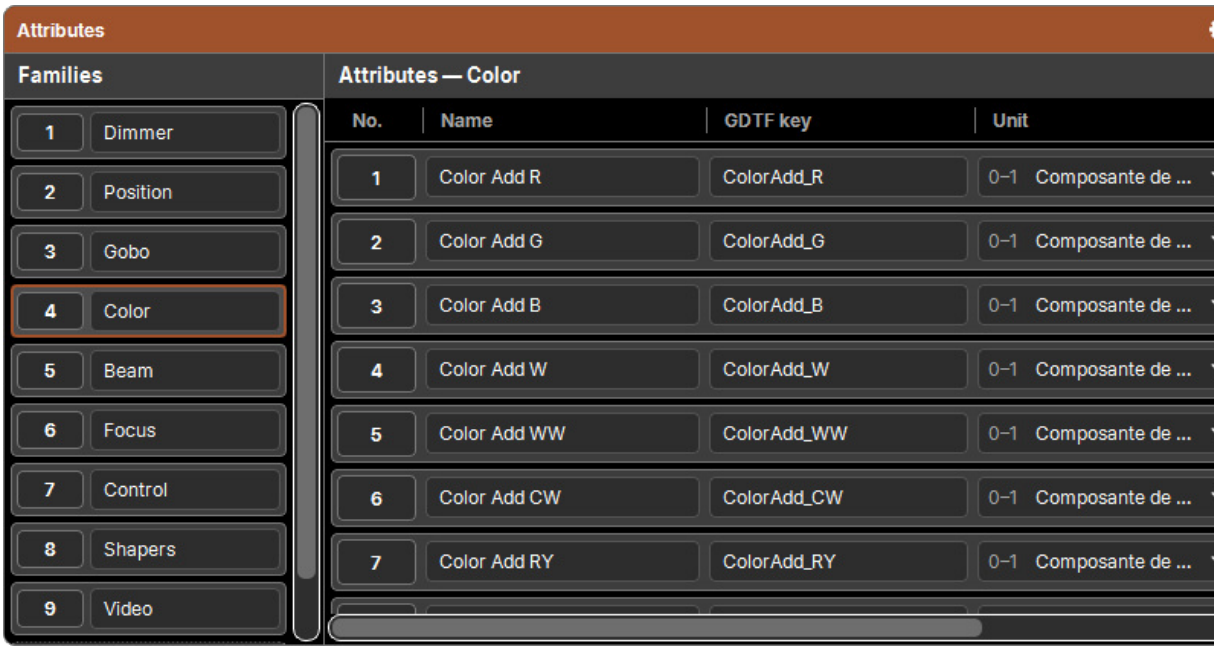
The computer's library

Families and attributes are kept in the computer's **library**, not in the show: they serve all your shows.

- The library always contains **the whole GDTF catalog**: the official families and the roughly 270 attributes of the standard (DIN SPEC 15800), with their key, name and unit. They are sorted in a logical order (Pan, Tilt before their settings; Red, Green, Blue, White before the variants), the same everywhere in the software.
- An official attribute that you delete **comes back at the next start**: the catalog is completed at every startup.
- Your own attributes are placed after the official ones of their family, in the order they were added.
- A show keeps a copy of the attributes used by its profiles. Opened on another computer, it finds them in that computer's library (by GDTF key, otherwise by name in the same family); missing ones are added.

Official family	GDTF key
Dimmer	Dimmer
Position	Position
Gobo	Gobo
Color	Color
Beam	Beam
Focus	Focus
Control	Control
Shapers	Shapers
Video	Video

The Attributes window



On the left, the list of **Families**; on the right, the **Attributes** of the family shown, as a table. A tap on a family (with no pad action) shows it. The family shown and the column widths are remembered with the window.

Column	Purpose
No.	the attribute's place in its family; it is the pad object (Delete)
Name	the name shown everywhere; typed directly
GDTF key	the attribute's name in the GDTF standard (`Pan`, `ColorAdd_R`, `Gobo1`...): it is what matches the attribute to GDTF files; empty ("custom") for an attribute of your own
Unit	the physical unit of its values (fixed list, see below)
Wheel	**No**, **Color** or **Image**: what its predefined values carry
Base value	a value that replaces the profiles' default for all fixtures; empty: each profile's default

Units

Symbol	Unit	GDTF key
—	None	None
%	Percentage	Percent
°	Angle	Angle
°/s	Angular speed	AngularSpeed
Hz	Frequency	Frequency
s	Time	Time
K	Temperature	Temperature
m	Length	Length

nm	Wavelength	WaveLength
cd	Luminous intensity	LuminousIntensity
0–1	Color component	ColorComponent

Wheel

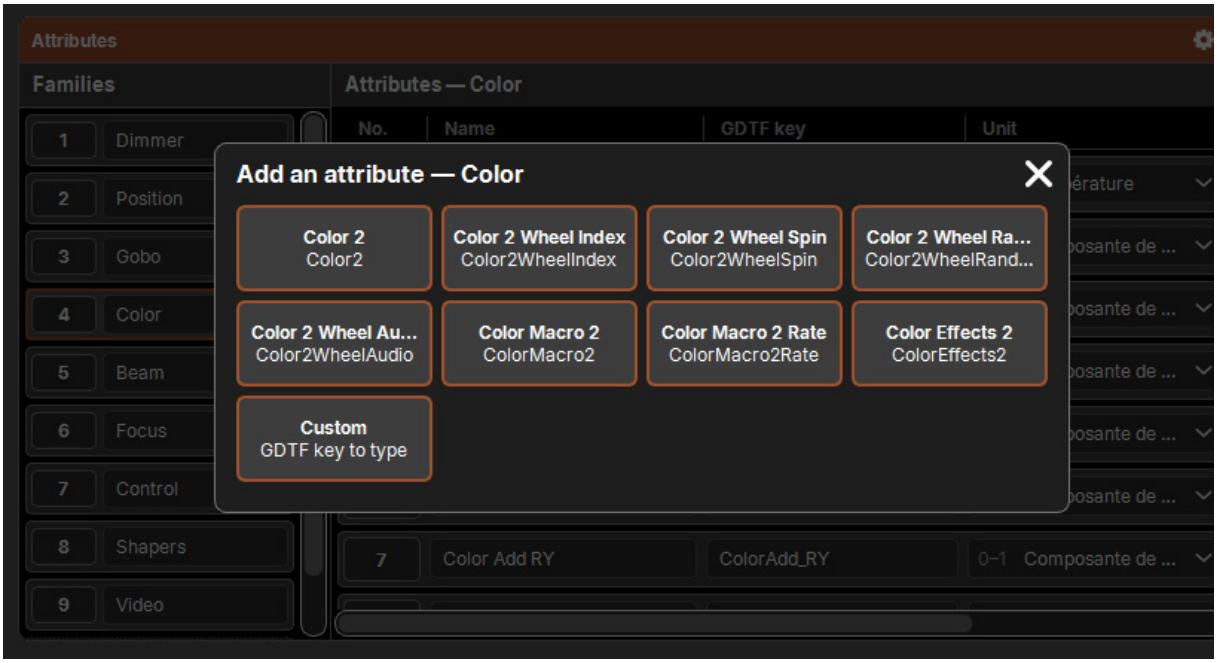
- **Color**: the attribute's predefined values carry a color (color wheel). In profiles, each value gets a color swatch; in the Programmer, the value tiles show that color.
- **Image**: the values carry an image (gobo wheel, animation wheel). Profiles and the Programmer show the thumbnail.
- **No**: nothing drawn, only the value's name.

Base value

The base value forces a default on **all** fixtures that have the attribute, whatever their profile. The common example: a **Dimmer at 0**, so that no fixture lights up by itself when patched (virtual dimmers included).

- It is typed in the unit shown: in % for an attribute in %, in degrees for an angle (`50`, `50 %`, `-90°`, comma or point).
- Clearing the field goes back to each profile's default.

Adding a family or an attribute



- **+ New family**, at the bottom of the family list, creates a family of your own. Rename it in its field.
- **+ Add an attribute**, at the bottom of the table, opens a grid **Add an attribute — Family**:
  - the **official** attributes of the family that are not there yet, with their name, key and unit filled in; for a numbered attribute, the first free number (**Gobo 2** if **Gobo 1** exists);

- **Custom**: a **New attribute** with no key, to name and set up yourself.

The added attribute is brought into view, ready to be renamed.

Renaming, editing, deleting

- Name, GDTF key and base value are typed in their cell; unit and wheel are picked from their list. A refused value goes back to the previous one.
- An attribute always keeps a name: an empty name is refused.
- **Delete**: **Delete** then a tap on the number of a family (it goes **with all its attributes**) or of an attribute. With a selection, the whole selection is deleted (see lists).
- No Copy or Move: families and attributes are created here, in catalog order.

All changes go through the history: **Undo** reverts them.

Profile-specific attributes

A GDTF file can contain attributes that are not in the standard ("Pan Invert", "LED Frequency"...). They do not clutter the library: they stay **specific to their profile**, with the family given by the file, and are shown in that profile only. The show keeps their list for its patched fixtures.

Names in English

The English interface shows, for an official attribute, its **GDTF name as separate words** ("PanRotate" → **Pan Rotate**, "Gobo1WheelSpin" → **Gobo 1 Wheel Spin**). An attribute you have renamed keeps your name in both languages. Nothing is changed in the library: you can switch from one language to the other without losing anything.

Profiles

A **profile** describes a fixture model: its DMX modes, its channels, what each value range does. It is the "fixture type" of grandMA and GDTF. Fixtures are patched from a profile (Patch); a profile is built by hand, imported from a GDTF file or downloaded from GDTF Share (GDTF and profile exchange).

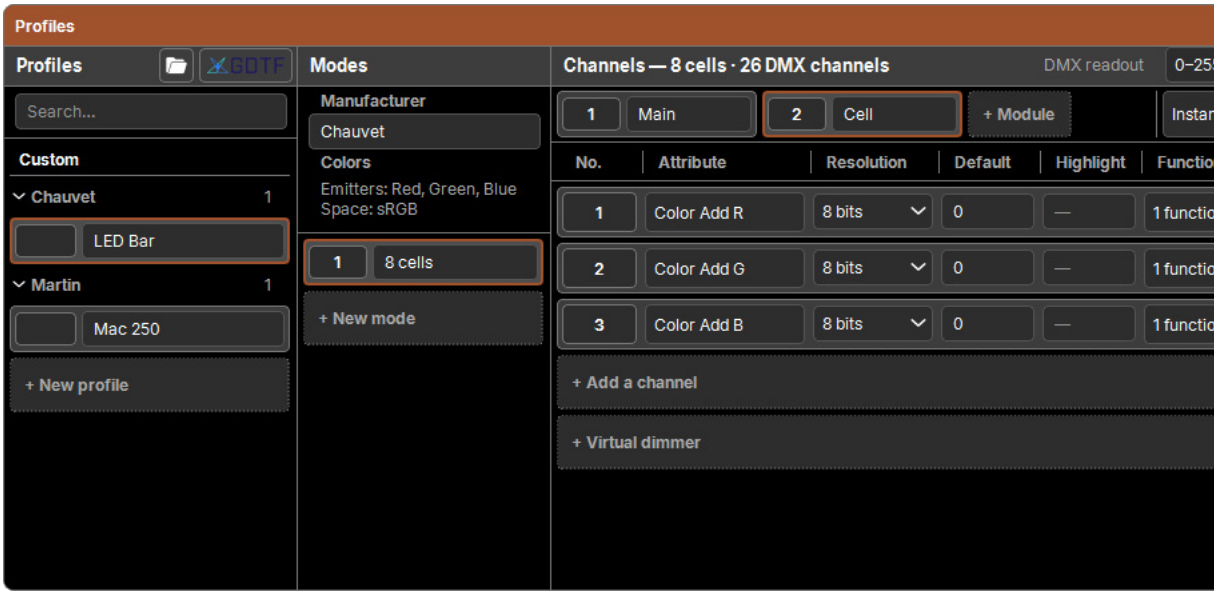
The **Profiles** window ("Add a window" → **Setup**) is the profile editor.

What a profile contains

Level	What it is	Example
Profile	the model, with its manufacturer	Robe Robin 600
Mode	one way of addressing it in DMX	Mode 1, 16 bit, Extended
Module	a pattern of channels (attributes in DMX order)	"Main": Pan, Tilt, Dimmer; "Cell": Red, Green, Blue
Instance	a module placed at an address of the mode	an LED bar places the "Cell" module 8 times
Channel	an attribute on 8, 16 or 24 bits	Pan in 16 bit
Function	a DMX range of the channel, with its attribute	0–9: closed, 10–255: Strobe
Predefined value	a name for a range within a function	"Open", "Gobo 3", "Red"

A simple fixture has a single module ("Main") placed once. The instances of a repeated module become **sub-fixtures** in the patch (101.1, 101.2...).

The Profiles window



Three columns, from left to right, each resizable by the handle on its edge:

1. **Profiles**: the search, then the list of profiles;
2. **Manufacturer**, colors and **Modes** of the profile shown;
3. the **modules** of the mode shown (as tabs) and the **Channels** of the module shown.

The profile, mode and module shown, the expanded brands and the widths are remembered with the window.

The profiles column

- **Search…** searches brand and name as soon as you stop typing; the brands with results are shown with their count.
- Two sections: **Custom** (your profiles, then **+ New profile**) and **Base** (the fixtures shipped with the software).
- Profiles are sorted by **brand**, collapsed: a tap on a brand (arrow, number of profiles) expands or collapses it. Profiles with no manufacturer are sorted under **No manufacturer**, at the end.
- A tap on a profile shows it (with no pad action).
- At the end of the header: **Export / Import** and **GDTF Share** (see GDTF and profile exchange).

The list stays fast even with thousands of profiles: only expanded brands have cards.

Base fixtures and custom fixtures

- **Base fixtures** ship with the software (a `Fixtures de base.elsf` file) and are updated with it. They cannot be deleted.
- Editing a base fixture breaks nothing: on the first change, the software makes a **custom copy** that replaces it everywhere (the history calls it "base → custom"). The base fixture itself keeps its content.
- **+ New profile** creates "Profile N" with a "Mode 1".
- **Copy** then a tap on a profile makes a custom copy ("… (copy)"), shown at once; with a selection of profiles, all of them are copied.
- **Delete** then a tap on a profile deletes it (except a base fixture).
- The name is typed directly in the card.

A show keeps its own copy of its fixtures' profiles: editing or deleting a profile in the library does not change a show that is already patched, which stays portable from one computer to another.

Manufacturer, colors and modes

- **Manufacturer**: typed in its field; it files the profile under its brand.
- **Colors** (if any): the emitters, filters and color space read from a GDTF file. For information.
- **Modes**: number and name. A tap shows the mode. **+ New mode** adds one ("Mode N"). **Copy** copies a mode at the end; **Delete** deletes it (a profile keeps at least one mode).

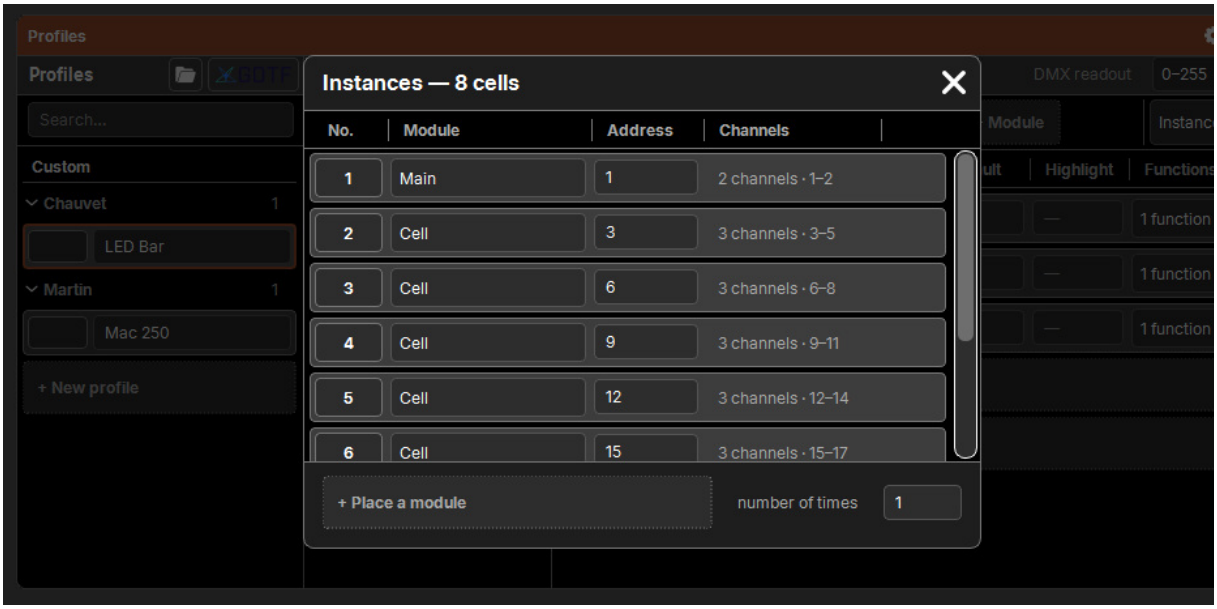
Modules and instances

Above the channel table, the **module tabs** of the mode: number (button) and name

(editable). They scroll with a drag.

- **+ Module** adds an empty module ("Module N").
- **Delete** then a tap on a module's number deletes it with its instances.
- **Instances (N)**, on the right, opens the **Instances — mode** popup.

The Instances popup

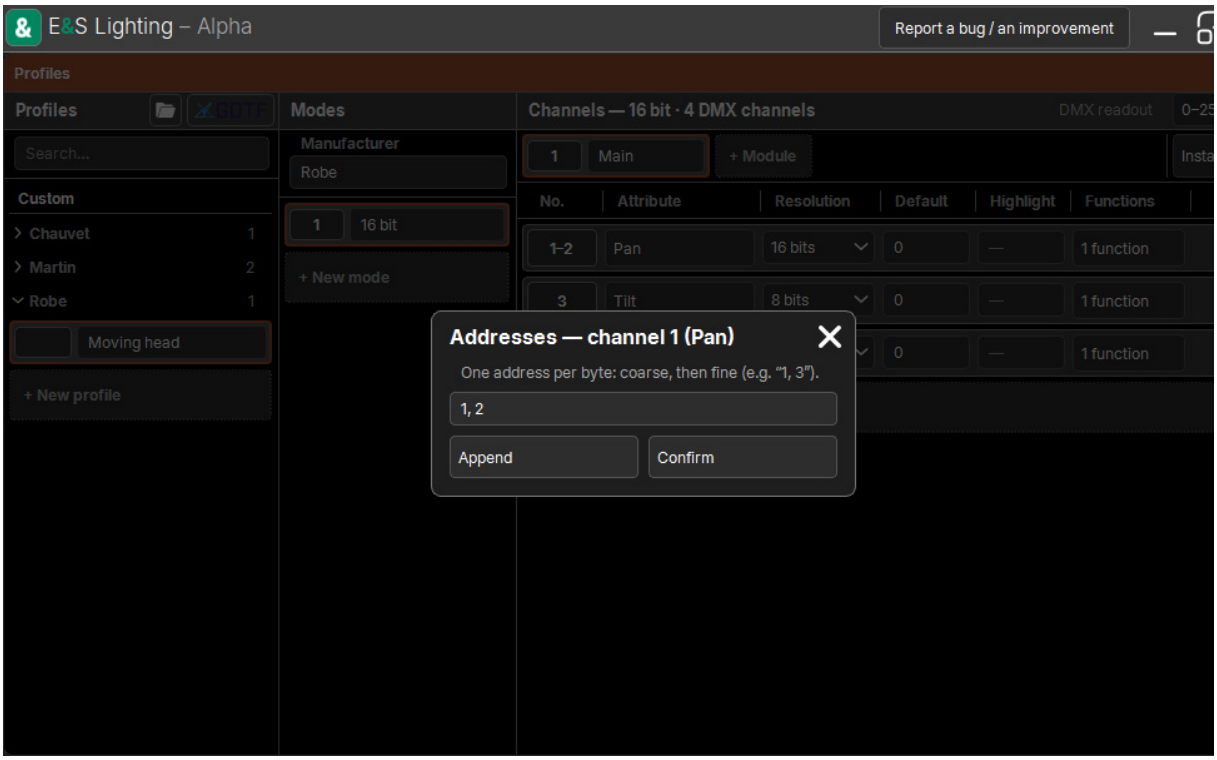


The instances of the mode, in order: this is the order of the future sub-fixtures.

Column	Purpose
No.	the instance's rank (pad object)
Module	the module placed; a tap changes it (for the whole selection if the cell is part of it)
Address	its first address in the mode; refused if it overlaps another instance
Channels	the channels it occupies

- At the bottom: **+ Place a module** and **number of times**: the chosen module is placed N times, after what is already occupied. Refused if the mode would exceed a universe.
- **Delete** on a number removes the instance (one instance always remains).

Channels



The channels of the module shown, in DMX order, with their addresses relative to the module.

Column	Purpose
Address	the channel button: "3", "3–4" for a 16-bit channel, "1 + 3" if the fine byte is elsewhere, "virtual" for a channel with no address
Attribute	a tap opens the choice families → attributes
Resolution	8, 16 or 24 bit
Default	the resting value, in the chosen DMX readout
Highlight	the value the attribute takes in Highlight; empty: none
Functions	the number of functions; a tap opens their popup

- **+ Add a channel** opens the choice families → attributes, without those already in the module: **an attribute appears only once per module**.
- A tap on a channel's address (with no pad action) opens the **Addresses** popup: one address per byte (coarse, then fine, then ultra: "1, 3" for a separate fine byte). **Append** makes them automatic (after the previous channel), **Confirm** saves. An address already taken or outside the module is refused, and the message shows in the command line.
- **Delete** then a tap on the address deletes the channel (the whole selection if it is part of it).
- **Move** then the channel, then another channel: it takes its place, the ones below move down. A tap on **+ Add a channel** sends it to the end. Move acts on the tapped channel alone, never on the selection. No Copy: an attribute only once per module.

- **Address**, **Resolution** and **Default** can be selected (a rectangle, or the whole column by its title): changing a selected cell changes the whole selection.
- Changing the resolution is refused if functions or values would merge.

Virtual dimmer

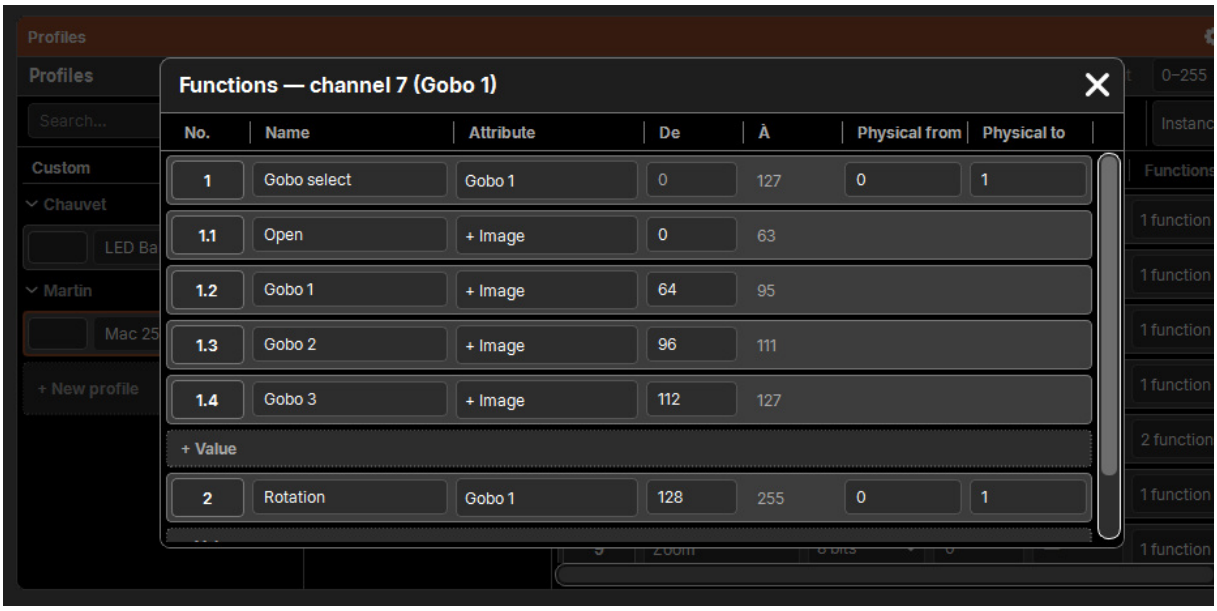
An LED cell often has red, green and blue, but no Dimmer. **+ Virtual dimmer** adds to the module a Dimmer **with no address** (shown "virtual") that **multiplies its colors**: you set the cell's intensity like any other fixture's, and the output computes the colors. It sends nothing in DMX.

DMX readout

Above the table, **DMX readout** chooses how DMX values are written: **0–255**, **0–65535** or **%**. It is a setting of the computer, like grandMA's "DMX Readout".

- In **0–255**, a 16- or 24-bit channel is written with one decimal (127.5) and its maximum is 255.
- Typing follows the same readout (comma or point, "%" optional).

Functions and predefined values



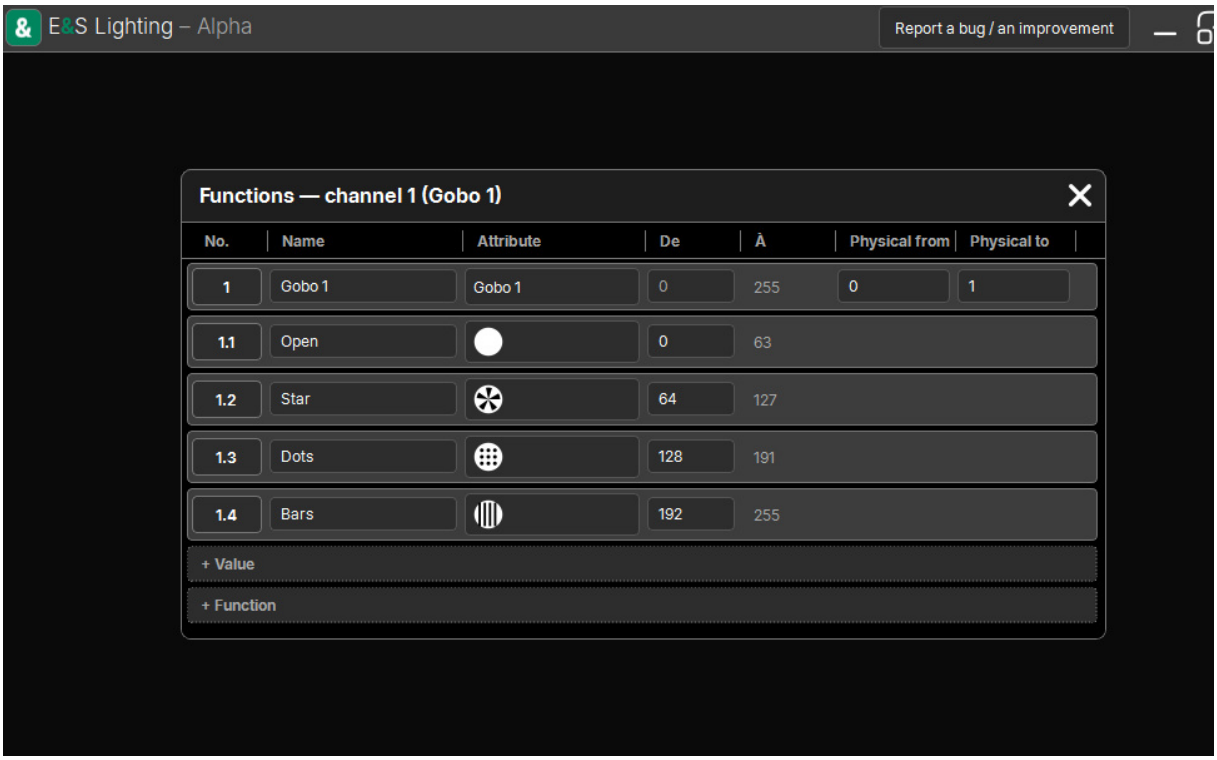
A tap on **Functions** opens the **Functions — channel** popup: each function (number "2") followed by its predefined values (numbers "2.1", "2.2"...).

Column	Purpose
No.	pad object (Delete)
Name	"Open", "Strobe", "Gobo 3"...
Attribute	the function's attribute (often the channel's, sometimes another one: a Shutter channel with a Strobe function); for a value, its color or image instead
From	the start of the range (the only field to type)
To	the end, computed: the range stops where the next

	one begins
Physical from, **Physical to**	the function's physical range in the attribute's unit (e.g. −270 to 270 °)

- Ranges **always follow each other**: the first function starts at 0, and you only type the starts. A start outside the function or the neighboring values is refused.
- **+ Function** splits the last function in two; **+ Value** splits the function's last value ("Function N", "Value N").
- **Delete** on a number: the neighbor takes over its range. A channel keeps at least one function.
- The number, the name and the attribute can be selected (a title: the whole column); changing a selected cell changes the whole selection.

Value colors and images



Depending on the **wheel** of the function's attribute (Attributes window):

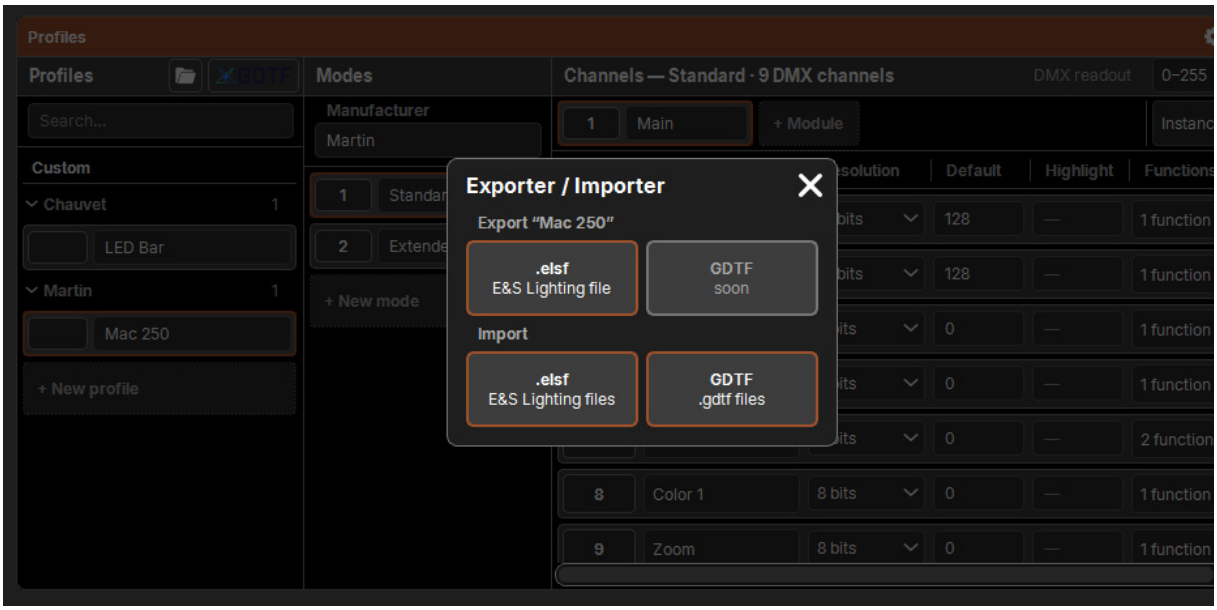
- **Color**: the cell shows the value's color swatch (split for a dual color), or **+ Color**. A tap opens **Color of the value**, where you choose one or more.
- **Image**: the cell shows the thumbnail, or **+ Image**. A tap picks an image file (**Image of the value**); **Delete** then a tap on the image removes it. Images go with the profile, reduced to thumbnails.

The popup follows **Undo** and **Redo** while it is open, and closes if the channel disappears.

GDTF and profile exchange

Profiles are shared in three ways: **GDTF** files (`.gdtf`, the manufacturers' open format), the **GDTF Share** online library, and E&S Lighting **.elsf** files (to move your profiles from one computer to another). Everything is done from the Profiles window, with the two buttons at the end of the profiles column header.

Export / Import



The **Export / Import** button opens a popup with two categories.

Export

- **.elsf** (E&S Lighting file): saves the **selected profiles**, otherwise the profile shown. The file manager opens in `Documents\E&S Lighting\Profils`.
- A base fixture you have edited is exported in its custom form.
- For many profiles, a popup follows the export ("Export: 120 / 3320"); it can be canceled.
- The result is shown in the command line: "12 profiles exported to ...".
- Export to GDTF is not available yet (grayed-out tile).

An `.elsf` file contains the profiles and the families and attributes they use: it opens as is on another computer.

Import

- **.elsf** or **GDTF**: choose one or more files in the file manager. Imported profiles become **custom fixtures**.
- With more than one file, a popup follows the import: reading the files, then saving. While reading, **Cancel** (or the cross) stops everything: nothing is imported.
- The first imported profile is shown; the summary is written in the command line: profiles imported, already up to date, old versions kept, errors (unreadable file, unreadable profile...).

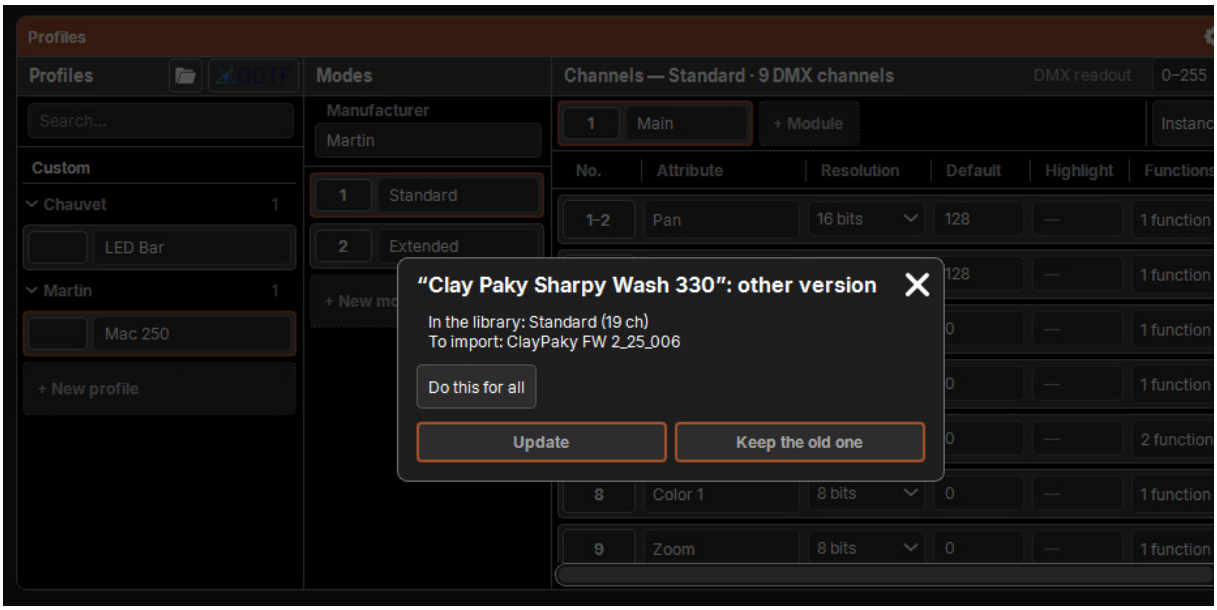
- The whole import is **a single** history **entry**: **Undo** reverts all of it, added families and attributes included.

Duplicates and versions

The software compares each imported profile with the library **before changing anything**; it only saves after the last answer.

Case	What happens
Same GDTF ID, same revision	skipped: already up to date
Same GDTF ID, other revision (or unknown revision)	**other version** question (see below)
The same profile twice in the import	error, the second one is ignored
No GDTF ID, but same brand and same name	"... already exists" question: **Replace** (only a custom fixture), **Import** (the name gets "1", "2"...); the cross cancels the whole import

Other version



The **"Name": other version** popup shows both revisions (**In the library** and **To import**), then:

- **Update**: the new revision replaces the custom fixture; a base fixture receives its update, filed under **Base** in its place;
- **Keep the old one**: nothing changes for this profile;
- **Do this for all**: the answer applies to the next ones in this import;
- the cross abandons the import.

What is read from a GDTF file

A `.gdtf` file is read and turned into a profile, with the same rules as `.elsf` files:

In the GDTF	In the profile

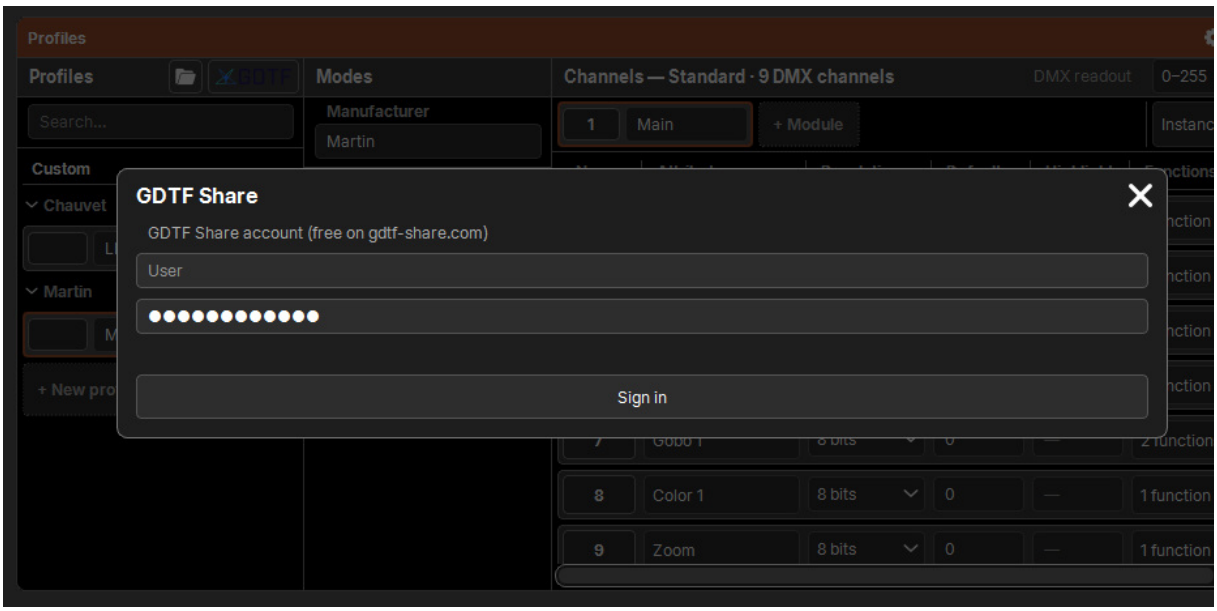
Manufacturer, Name, FixtureTypeID	manufacturer, name, GDTF ID
Official attributes	the library's attributes, in their official family
Other attributes	attributes specific to the profile
DMXMode	a mode
DMXChannel	a channel (its attribute, its bytes, its addresses, its default)
ChannelFunction	a function (its start, its attribute)
Named ChannelSet	a predefined value
Repeated geometries (Head_1 to Head_12...)	a "Cell" module placed once per head: sub-fixtures in the patch
Channel with no address that multiplies its geometry	a virtual dimmer
Emitters, filters, color space	the profile's **Colors** (information)

Not read yet: the second DMX break (DMXBreak 2...), wheels, physical ranges. What is not read is reported in the import summary.

GDTF Share

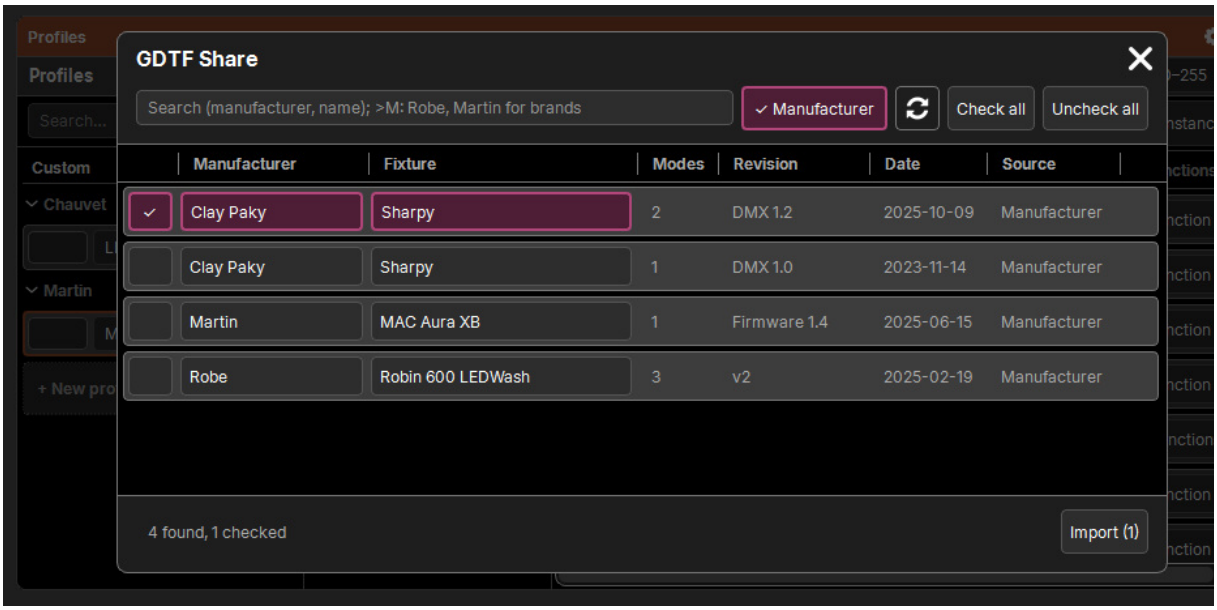
GDTF Share (gdtf-share.com) is the online library of GDTF files, fed by manufacturers and users. You need an account, free, on their site.

Signing in



The **GDTF Share** button opens the popup. Enter your GDTF Share **User** and **Password**, then **Sign in**. The user is remembered by the computer; the password goes to the system vault (the Windows Credential Manager), never into a database.

Choosing and importing



Once signed in, the list of all GDTF Share fixtures loads (it is kept for the session; the arrows button, **Refresh the list**, reads it again).

- **Search**: in the manufacturer or the name. `>M: Robe, Martin` restricts to brands ("Robe" also finds "Robe Lighting").
- **Manufacturer** (checked at first): only the files uploaded by the manufacturer itself, the most reliable.
- One card per **revision**, from newest to oldest: manufacturer, fixture, modes, revision, date, source (Manufacturer or User). Check the ones to import.
- **Check all** / **Uncheck all**: what the search shows.
- **Import (N)**:
    1. the checked revisions are compared with the library **before any download**: those already up to date are skipped, another version asks **Update / Keep the old one**; a fixture checked twice, only the newest counts;
    2. only new ones and updates are downloaded ("Download: 12 / 40"); the button becomes **Cancel**;
    3. then imported, like files.

Downloaded files stay in `Documents\E&S Lighting\GDTF`. Closing the popup while signing in or downloading abandons the rest.

Patch

The **patch** says which fixtures the show contains: each one has a **number**, a name, a profile and a mode, and a **DMX address**. Fixtures are then selected by their number on the pad, in groups, cues and presets.

The **Patch** window ("Add a window" → **Setup**): the **sections** on the left, the fixtures of the section shown on the right. The section shown and the column widths are remembered with the window. Thousands of fixtures stay smooth: only the cards on screen exist.

Addresses and universes

An address is written **universe.address**: `1.1`, `2.17`, `12.301`.

- A number alone is read as **absolute**: `513` is address `2.1` (universe 2, address 1).
- A universe has 512 channels; you can go up to **63,999 universes**, as in sACN. The software limits nothing: it only says when the computer cannot keep up (see DMX Output).
- A fixture **never** spans two universes: one that no longer fits in a universe moves to the start of the next one.
- An empty address: the fixture has **no address** (unpatched). It stays in the show, can be programmed and is shown in Layouts, but outputs nothing.

Only universes where a fixture is patched are sent.

Adding fixtures



+ Add a fixture, at the bottom of the table, opens a two-step popup.

1. **Profile and mode**: the Profiles and Modes columns of the Profiles window (search, collapsible brands). Choose a mode, then **Next**.
2. **Where to put them**: the chosen profile and mode (with their channel count),

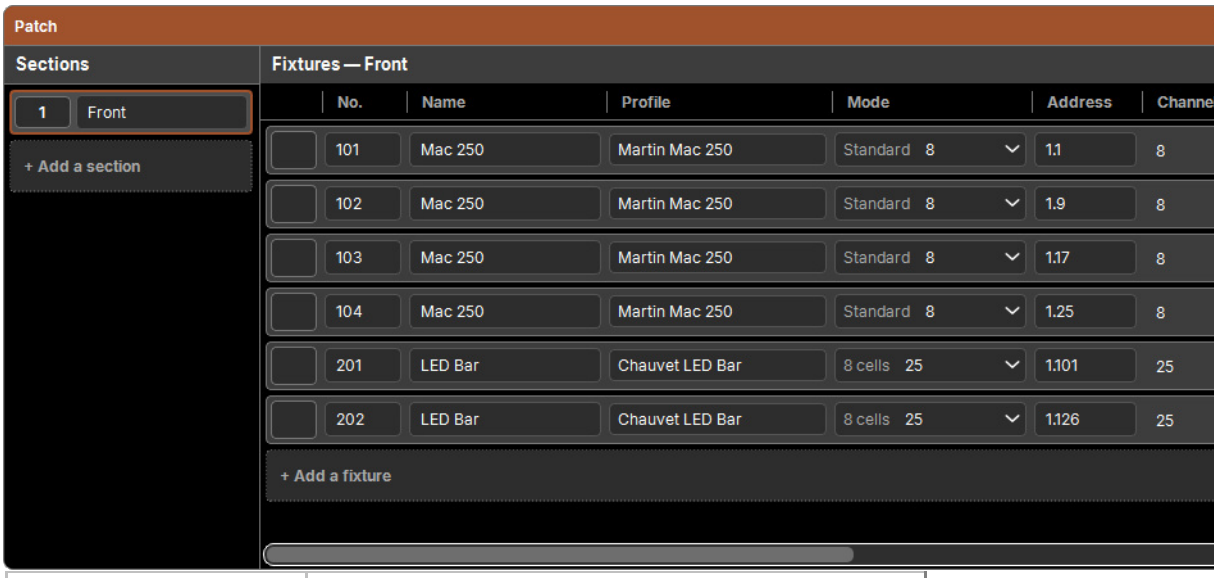
then:

Field	Purpose
Number	the number of the first fixture; suggested: the first free one after the section's fixtures
Count	how many fixtures (1 to 1000), numbered in sequence
Address	`universe.address` of the first one; empty: no address; suggested: the first place where they all fit without overlapping anything
Gap	free channels left between two fixtures
Name	the fixtures' name; suggested: the profile's name

Everything is checked as you type: what is wrong is written in red (number taken, address overlapping a fixture, outside the universe…) and **Patch** stays grayed out. **‹ Back** returns to step 1.

Patch creates the fixtures in the section shown, as a single history entry. The profile enters the show (the show keeps its own copy of its fixtures' profiles: it stays portable).

The fixture table



Column	Purpose
(button)	the pad object: **Delete**, **Copy**, **Move**
No.	the number; can be changed if it is free
Name	typed directly
Profile	a tap chooses another profile
Mode	the list of the profile's modes
Address	`universe.address`
Channels	the channels occupied ("1.1–1.16")

Last channel	the end of the fixture
Sub-fixtures	the instances of a repeated module (see below)
Offset Pan, **Offset Tilt**	an offset in degrees
Invert Pan, **Invert Tilt**	**Yes** / **No**: the DMX output inverted

A refused change (number taken, overlapping address, mode too large) goes back to the previous value, and the reason is written in the command line.

Changing profile or mode

- **Mode**: the list of the profile's modes. For a selection of fixtures of other profiles, each one takes the mode **with the same name** (with no mode of that name, it does not change).
- **Profile**: a tap opens the choice of profiles and modes, then **Replace**. The fixtures keep their number and address; a name still equal to the old profile's takes the new one's.
- In both cases, it is refused if the new size overlaps another fixture or exceeds the universe.

Sub-fixtures

A fixture whose profile places a module several times (an 8-cell LED bar, a multi-head fixture) has **sub-fixtures**: fixture **101** has sub-fixtures **101.1** to **101.8**.

- **101** means the whole fixture (its head and all its cells);
- **101.3** means the third cell alone, wherever fixtures are designated: selection, groups, cues, presets;
- **101.0** means the head alone, without its cells (its square in a Layout).

Selection and sequential entry

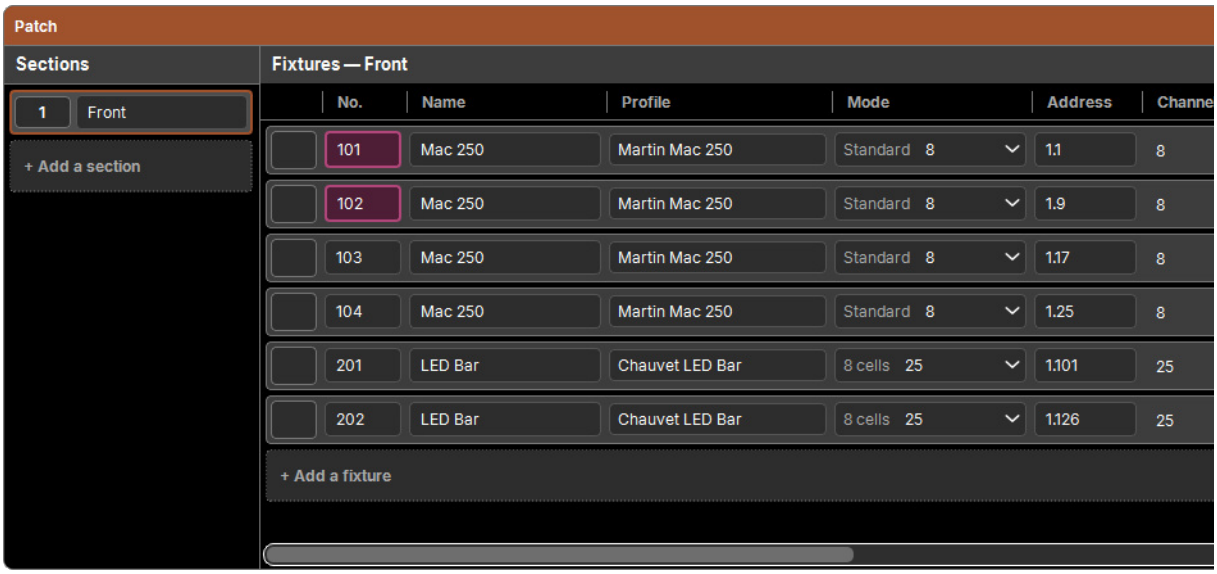


Table cells can be selected: a rectangle with a finger or the mouse, or a whole column by tapping its title (see selection). Then:

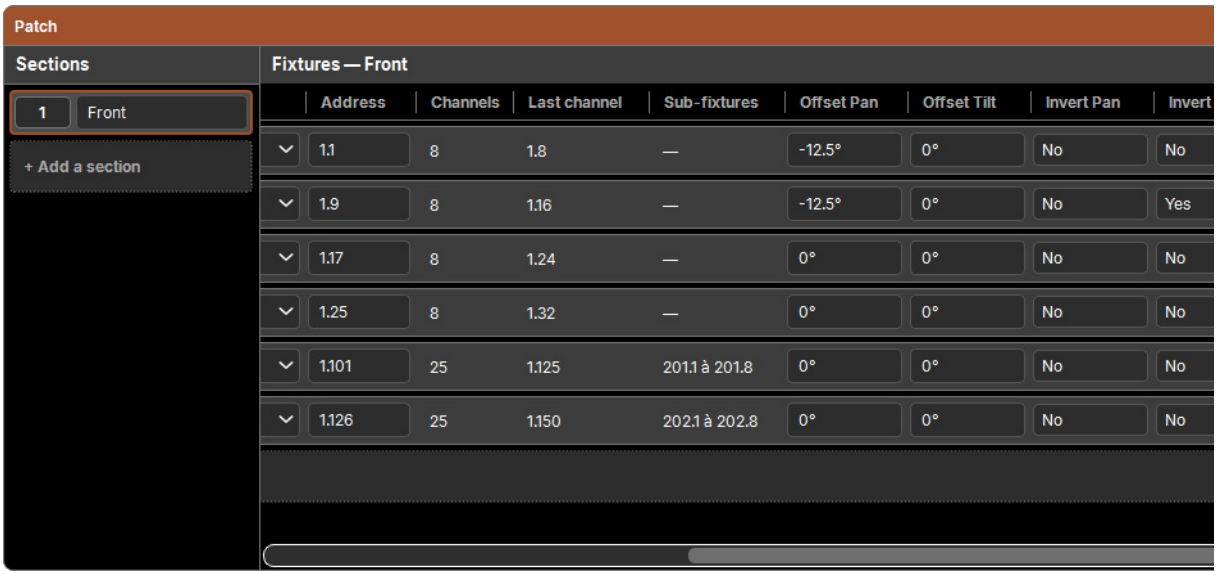
- **No.** selected: type a number in the command line, then **Enter**: the fixtures are

renumbered in sequence, in the order you selected them. A number already taken by another fixture refuses everything.

- **Address** selected: type an address, then **Enter**: they are **patched in sequence** from that address. Type **.** alone to **unpatch** them all.
- **Profile** or **Mode** selected: the choice made in one applies to all.
- **Offset** selected: the typed value sets them all; **Invert** selected: a tap on one toggles them all.

This is the fast way to repatch a whole lighting plot: select the addresses in the desired order, type the first one, **Enter**.

Rigging: Offset and Invert



As in grandMA's patch, each fixture has its **rigging**:

- **Offset Pan** and **Offset Tilt**: an offset from −360 to 360° ("-12.5", "10°"), so that the same position aims at the same point whichever way the fixture is hung;
- **Invert Pan** and **Invert Tilt**: the DMX output inverted (a fixture hung upside down, or the other way round).

Rigging is applied **just before the output**: the Programmer, cues and presets do not see it. A position preset therefore stays valid for all fixtures, whichever way they face.

Delete, Copy, Move

The procedure is the same as everywhere (see lists):

- **Delete** then a tap on a fixture's button deletes it (with the selection if it is part of it).
- **Copy** then the fixture, then another fixture (or **Enter**: after the last one): copies arrive **with no address**, to be repatched, so as not to overlap anything by surprise.
- **Move renumbers**: the source takes the target's numbers (**Replace**, **Append**). Numbers are those of the whole patch, not of a section.

Sections

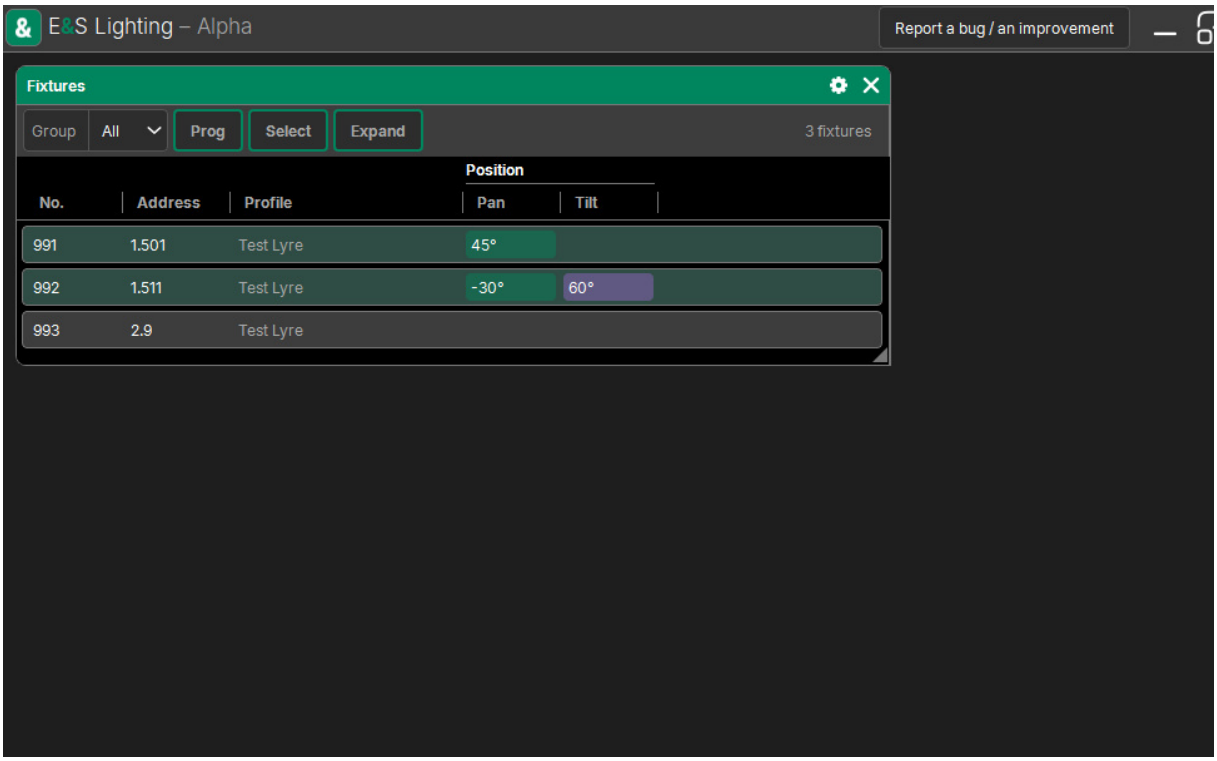
Sections organize the patch (by truss, by type, by area…); they change nothing at the output.

- **+ Add a section** creates one, which is shown; rename it in its field.
- A tap on a section shows it; added fixtures go into the section shown.
- **Delete** then a tap on a section deletes it **with its fixtures** (Undo brings everything back).

The Fixtures table

The **Fixtures** window ("Add a window" → **Programming**) shows **all the fixtures in the patch** and, for each one, **what is actually output** on each active attribute. It is the equivalent of grandMA's Fixture Sheet: you see at a glance who does what, and where each value comes from.

What it shows



- On the left: the fixture's **No.**, **Address** and **Profile**.
- Then **one column per active attribute**, sorted under the name of their family (Dimmer, Position, Color...), with a line under each family.
- An attribute is **active** when something is driving it right now: the Programmer, a Cue List in playback, an effect. Highlight does not count.
- The value shown is **the one that is output** (after priorities, fades, masters), updated continuously.
- The **background** of a cell takes the color of what drives it: the programmer, a Cue List, an effect; a parked fixture has its cells in blue, with a band under its number.

A column only appears if at least one fixture shown has that attribute active: the table stays readable even with hundreds of attributes in the show.

Sub-fixtures

A fixture that has sub-fixtures has an **arrow** in front of its number: a tap on this cell **expands** its instances below (101.1, 101.2...), indented, each with its values. Another tap collapses them.

The selection

Fixtures **selected** in the Programmer stand out (a tint in the programmer's color), their instances too. The table does not change the selection: it shows it.

The bar

At the top of the window:

Element	Purpose
Group	only show the fixtures of a group; **All** at first: the whole patch
Prog	only the fixtures active in the programmer
Select	only the selected fixtures
Expand	expand all fixtures with sub-fixtures

Sorting

A tap on the **No.**, **Address** or **Profile** title sorts the table on that column (ascending, descending, then the original order). An attribute column does not sort: its value changes every frame.

The group, the filters, the sorting and the column widths are remembered with the window.

Big shows

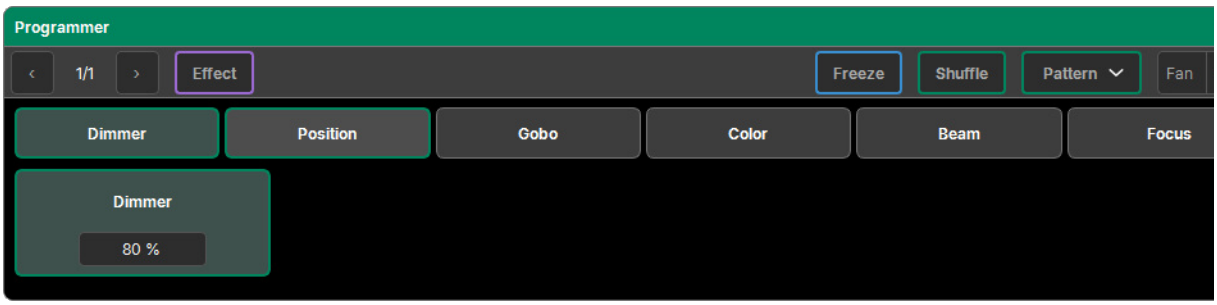
The table is built for thousands of fixtures: only the rows on screen exist and are watched at the output. Scrolling costs nothing to the light calculation.

The Programmer

The programmer is your workspace, just like on grandMA: you select fixtures, give them values and effects, then store the result in a cue, a preset or a group. Whatever is in the programmer **is output live**.

The programmer is **not saved in the show** and does not go through the history: it is a scratchpad. Only what you store is kept.

The Programmer window



"Add a window" → **Programming** → **Programmer**. It fits in three rows, with no scrolling at all:

1. **The tools**: on the left **‹ 1/2 ›**, the attribute pages of the family shown; then the **Effect** button (see Effects); on the right **Freeze**, **Shuffle**, **Pattern** and the **Fan** list.
2. **The families** of the patched fixtures (Dimmer, Position, Color, Gobo...), sharing the width.
3. **The attributes** of the chosen family, four per page, as tiles.

Families

- A tap chooses the family to show (green-gray background).
- A family that no fixture of the selection has is **grayed out** and cannot be chosen.
- A family with a value in the programmer is **outlined** in the programming color (darker outline if its values are inactive); **purple** if one of its attributes has an effect.
- A **double tap** on a family brings it into the programmer as it is currently output, like **On** (see below).
- The attribute page of each family is kept until Clear: coming back to a family finds it where you left it.

Attribute tiles

A tile shows the attribute **name** and, at the bottom, its **value**: the average of the selection. For a fixture that is not in the programmer, this is what is really output (the running Cue List, a fade in progress...), updated live.

- A set tile is outlined in the programming color, on a green-gray background; inactive (after Off or Clear), only the outline, darker.
- An attribute specific to a profile (not in the library) also shows each fixture's value under its number.

Selecting fixtures

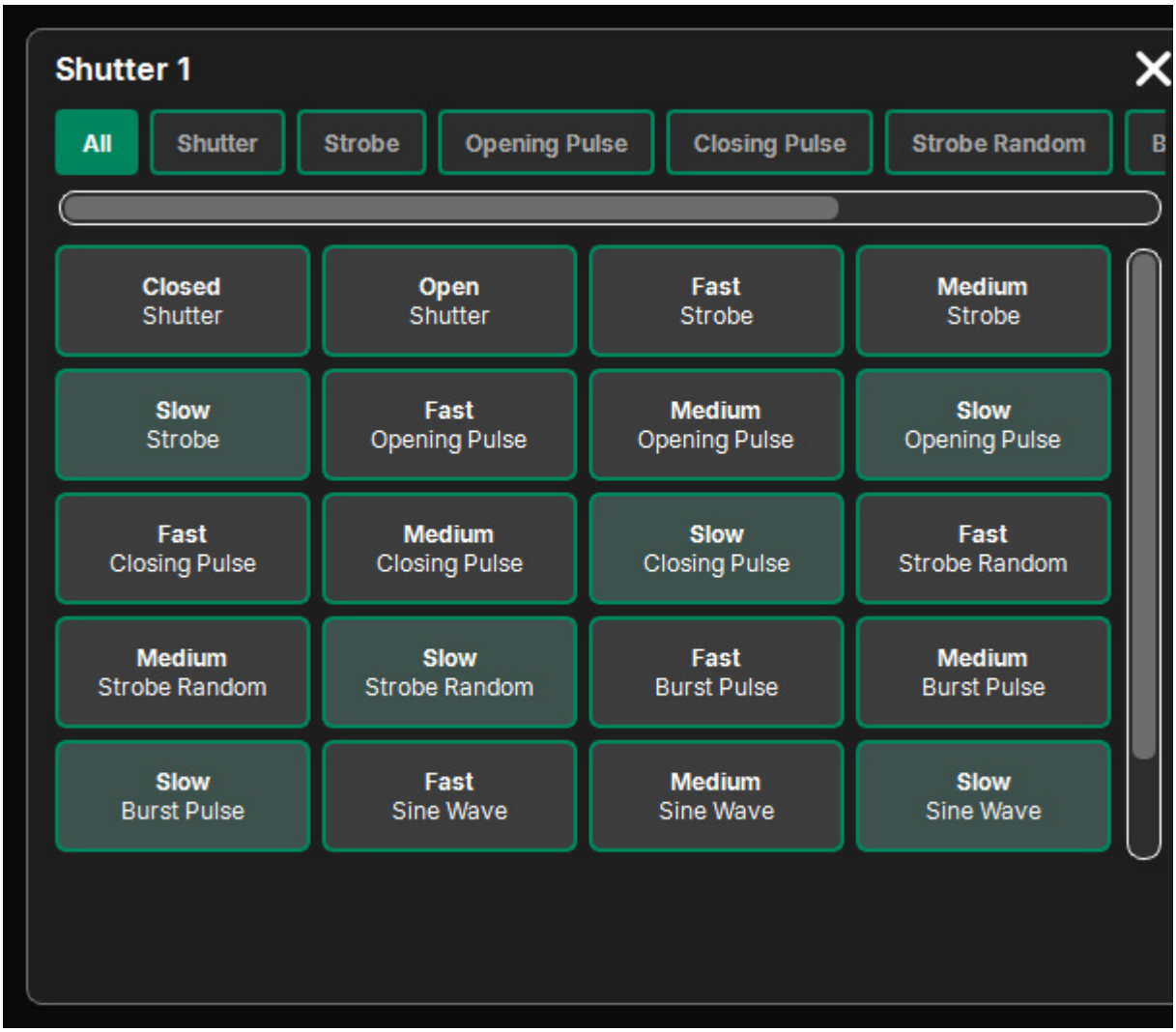
Fixtures are selected:

- from the keypad: **Fixture** then a number or a range, then **Enter**: "Fixture 1 **Thru** 50"; "50 Thru 1" selects in descending order; a sub-fixture is written "101.3", the head alone "101.0";
- with a tap on a **group** (see Groups);
- in a **Layout**, with a tap or by drawing a rectangle (see Layout);
- in the **Fixtures** window (see Fixtures).

The selection rule, as on grandMA: as long as no value has been changed, a new selection **adds** to the previous one. As soon as a value has changed, the next selection **starts from scratch**; the values already set stay in the programmer.

The **selection order** matters: it drives the spread of a value range, the Fan and the effect offsets. It is arranged in the Grid.

Giving a value



- **Typing**: a tap in a tile's field clears it, ready to type; **Enter** confirms. Left

without typing anything, it goes back to its value and nothing is sent. The keypad digits go into the field being edited.

- The value goes to **the whole selection**, each fixture in its own function and limited to its range.
- **A range** "0 **Thru** 100" is spread over the selection, in grid order (first fixture: 0, last: 100). With several points, "0 Thru 100 Thru 0" makes a bump. The direction is set with the **Fan** and the Pattern.
- **Predefined values**: a tap on the tile's **name** opens its named values as tiles (Open, Closed, Gobo 3, Strobe...). A channel with several functions (a Shutter: Shutter, Strobe, Pulse...) shows its functions as categories at the top, preceded by **All**. The engaged value is highlighted. The popup reopens on the last function chosen for this attribute, until Clear.

A single-function channel is set in the attribute's unit (Pan in °, Dimmer in %). A channel with several functions is set over **the whole channel**, from 0 to 100%: the value lands in the function that contains that DMX value.

Units, RGB or CMY

The gear of the Programmer window sets the display, without changing anything that is sent or stored:

- **Colors** → **Program in RGB** or **CMY**: the color tiles show Red, Green, Blue or Cyan, Magenta, Yellow; input follows, converted onto each fixture's real attributes.
- **Units**: for each attribute that has a physical unit, **Real value** (°, Hz, s...) or **%**.

On, Off and Release

These keypad keys act on the programmer when you then tap an **attribute** (its name) or a **family**, for the selected fixtures:

Key	Effect
On	the attribute enters the programmer at its **current value**, without the light moving (an inactive value becomes active again)
Off	its values **leave** the programmer: the light goes back to what is underneath (the Cue List, otherwise the default)
Release	the cell gets a **Release** value: nothing is output from the programmer; stored in a cue, it releases that cell of the Cue List

On whole fixtures:

- **Off** then **Enter**: the whole selection leaves the programmer (and the selection); "Off Fixture 5 **Enter**" too, or **Off** then a group, a fixture of a Layout or of the Grid.
- **Release** then **Enter**: all the attributes of the selection; "Release Fixture 1 Thru 5" too.

A quick double tap on **Off** or **On** gives **Park** or **Unpark**: see Park.

Highlight, Solo and stepping through

- **Highlight**: the selected fixtures output their **Highlight values** (those of the profile; failing that, Dimmer at full and Shutter open), above everything. A toggle: a second tap turns it off.

- **Solo**: the intensities of all **other** fixtures are output at 0. A toggle too.
- **<** and **>**: the selection is stepped through **fixture by fixture**; what you set only applies to the current one (handy for setting positions one by one). **Clear** goes back to the whole selection; another selection ends the stepping.
- **Set** (between **<** and **>**): a tap toggles between the whole selection and the sub-selection you came from. **Held** with **>** or **<**: one fixture in 2, in 3... (the Interleave of grandMA2). With a mouse, hold Set for a moment then release: it stays engaged until the next tap.

Highlight, Solo and stepping through go neither into the programmer nor into a cue.

The toolbar tools

Freeze

When lit (in ice blue), the programmer stays **above the Cue Lists**, even after a more recent Go (only a Cue List with Super priority still goes over it). It stays lit until you turn it off; Clear does not touch it.

Shuffle

The Shuffle of grandMA3: the selected fixtures **swap their grid cells** at random. Everything that reads the grid follows: value ranges, Fan, effects.

- Each tap moves to the next shuffle ("**Shuffle 3**", lit); the same number always gives the same shuffle.
- The small cross goes back to the original order.
- The draw is **spread out**: two fixtures that were neighbors before are not after, even across thousands of fixtures.

Fan

The Align of grandMA: the direction in which a range "a to b" is spread.

Symbol	Spread
-	no direction imposed (like <)
<	from the first fixture to the last
>	from the last to the first
><	a at the edges, b in the center
<>	a in the center, b at the edges

Clear does not touch it.

Pattern

It opens the Pattern panel, the MAtricks of grandMA3: see The Pattern.

The programmer's priority

The programmer outputs on a **"latest takes precedence"** basis against the Cue Lists, like grandMA:

- each value is timestamped when it is set: set after the Go of a Cue List, it goes above (intensities included);
- a cue that changes that value afterwards takes over again; a value that is only tracked does not;

- a Cue List with **Super** priority stays above;
- with the programmer emptied, the light goes back to the Cue Lists.

With **Freeze**, the programmer keeps control even after a more recent Go.

Clear

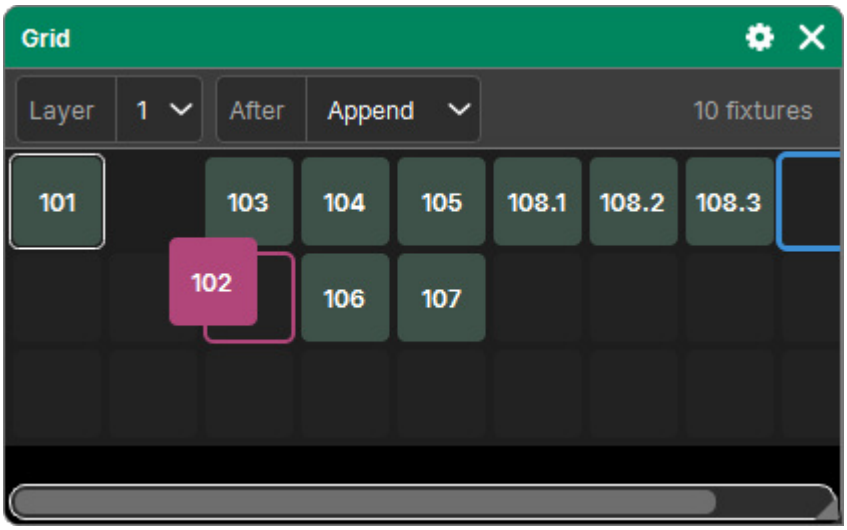
Clear only touches the programmer, in **three steps**, like grandMA:

1. if there is a selection (or a shuffle, a Pattern): the selection is released, then the Pattern;
2. otherwise, if there are active values: they are **deactivated**. They stay and the light does not move, but they will no longer go into a store;
3. otherwise: the programmer is **emptied**, effects included.

If you were stepping through the selection with **<** / **>**, the first Clear first goes back to the whole selection.

Esc is not Clear: it cancels the keypad action in progress and empties the selected cells of the tables, without touching the programmer.

The Grid



Each selected fixture has an x, y, z **cell**, like the Selection Grid of grandMA3. It gives the order of spreads, the Fan, the Pattern and effects.

- A selection made from the keypad or by tapping is placed **one after the other** on the cursor's row.
- A **group**, a **preset** or a **rectangle in a Layout** brings its own grid (in a Layout: one column per position, one row per height), placed at the cursor.
- A deselected fixture leaves its cell empty; a new selection (after a value changed, or Clear) starts again from cell 0, 0.
- Two fixtures at the same x, y place on different layers receive the same value.

The **Grid** window (Programming) shows the grid:

- the layer shown (**Layer** list), the fixtures at their number ("101", "101.3"), the origin in a white frame, the cursor in blue;
- a tap on a cell puts the **cursor** there;

- dragging a fixture puts it on another cell (the one that was there takes its place); dragging elsewhere scrolls;
- **Off** then a fixture removes it from the programmer;
- **After**: where the cursor goes after a selection: **Stay**, **Append** (default) or **New line**.

Storing

The **active** content of the programmer is stored with **Store** then the target:

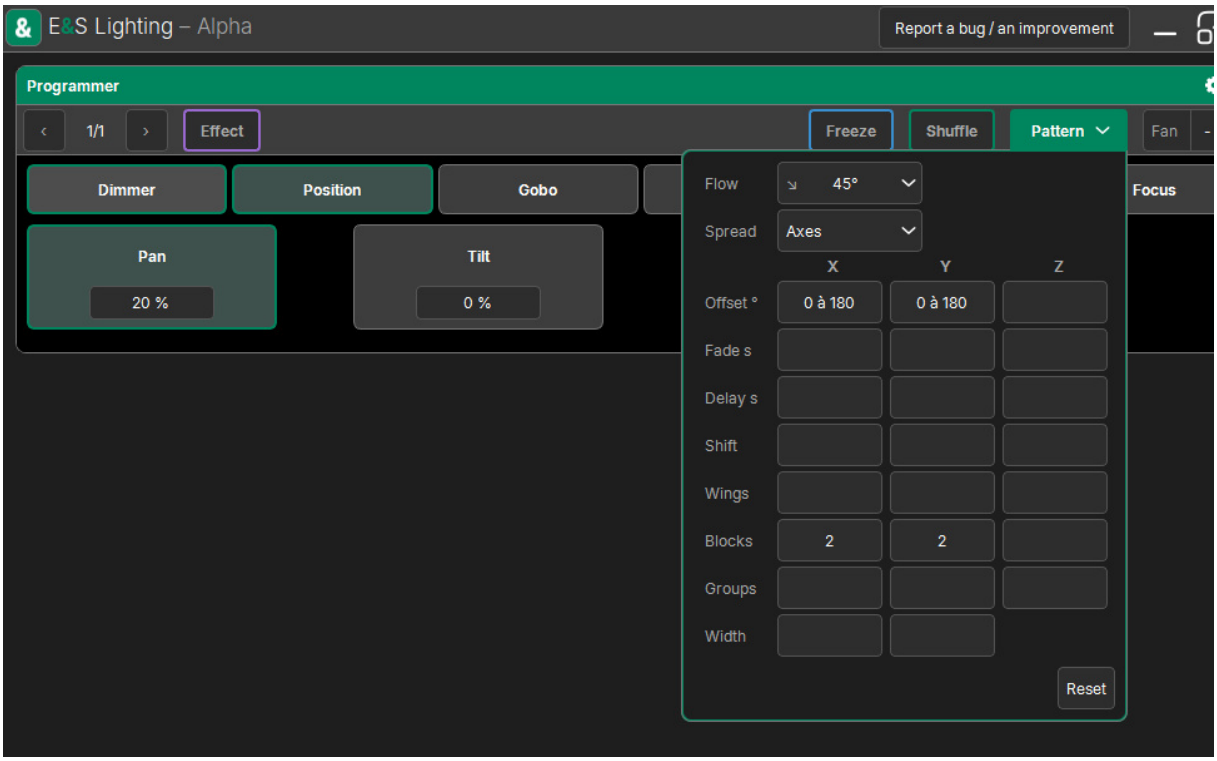
- a cue: see Storing cues;
- a preset (only the pool's family): see Presets;
- a group (the selection): see Groups.

A store also keeps each fixture's cell in the grid, and the Pattern times.

The Pattern

The Pattern is the equivalent of grandMA3's **MAtricks**: it defines how a value range, an effect offset or times are **distributed** over the selection. There is **only one Pattern**, shared by the selection and the effects: the effect editor is just a shortcut to it.

Opening the Pattern



In the **Programmer** window, the **Pattern** button (between **Shuffle** and **Fan**) opens a panel under it. The button is lit when a Pattern is set. A tap outside the panel, or **Esc**, closes it.

The panel shows the Pattern table, then **Reset**, which clears it.

Flow and Spread

At the top of the table:

- **Flow**, the shortcut: **Reading**, the **8 arrows** (→ left to right, ↓ top to bottom, ↘ diagonally...), **Center** (from the center to the edges), **Edges** (from the edges to the center), and the **3D** directions for volume shapes. "**Axis by axis**" is shown when the setting matches no shortcut. A direction chosen with no offset takes a full turn, "0 to 360".
- **Spread**: how the selection grid is traversed.

Spread	Traversal
Reading	a single counter, cell by cell, row by row (like grandMA2)
Axes	X, Y and Z each have their own setting; their

	offsets add up (like grandMA3)
Radius	from the center of the grid to the edges, in a circle; **Angle °** and **Center at** move the center (arcs)
Sphere, **Cube**, **Diamond**, **Cylinder**	volume shapes, from the center to the edges; **Center Z** sets the depth of the center

The table: one column per axis

One column per axis (**X**, **Y**, **Z**), one row per setting. An empty field: none. In **Reading** and **Radius**, only the X column counts (a single counter); **Width** always applies.

Setting	What it does
Offset °	the "from to" offset (the From-To Phase): "0 to 360" spreads a full turn along the axis
Fade s	a "from to" fade time per fixture, in seconds
Delay s	a "from to" delay per fixture, in seconds
Shift	the pattern slides by N cells (negative: the other way), looping
Wings	the axis split into N parts, each mirroring the previous one
Blocks	N consecutive cells act as one
Groups	the pattern repeated every N cells
Width	the row cut every N cells, continuing on the next row (X wraps onto Y, Y onto Z): to make a grid out of a simple row

Example: 12 fixtures in a line, **Offset** "0 to 360" and **Wings** 2 on X: an effect starts from both edges at the same time and meets in the center.

Value ranges follow the Pattern

A typed value "a **Thru** b" follows the same wave as an effect: the fixture that would start first gets **a**, the last one **b**. With the → direction a whole column gets the same value; with ↘ diagonally; with **Center**, from the center to the edges. With no offset set, the range is spread cell by cell with the X settings.

With several points, "0 Thru 100 Thru 0", the values are distributed at equal distances (a bump). The **Fan** keeps its direction on top.

Per-fixture fades and delays

Fade s and **Delay s** give each fixture its own time, spread like the offset (Shift, Wings, Blocks, Groups apply; in Axes, the axes add up).

- A value set in the programmer takes its fixture's time. In the programmer, it changes immediately: the time only matters for the cue.
- The cue **keeps this time with the value**, per fixture and per attribute. On Go, it takes precedence over the family time, which takes precedence over the cue time; the cue duration takes it into account.
- The Cue List shows the range of these times in its **Fade** and **Delay** columns.
- Changing the Pattern afterwards does not touch values already set.

The Pattern and effects

- Each effect line keeps a **copy** of the Pattern: a cue or a preset keeps the result.
- When the Pattern changes (the panel, an effect line, the Phase or the Flow of the band), **all the effects of the selected fixtures** take it at once, all families. Those of other fixtures and of cues keep their copy.
- A ready-made shape gives its own to the Pattern if it is empty (the Wave: 0 to 360).

See Effects.

What the Pattern is not

- It is **not saved in the show** and does not go through the history: it is a programmer setting.
- A new selection does not touch it; **Clear** releases it on the first step, with the selection (effects already set keep theirs).

Groups

A group is a **stored selection of fixtures**, in its order and with its grid, as on grandMA. It is used to reselect with a tap ("the front profiles", "the back washes"), to carry an intensity Master and to define a World.

The Groups window



"Add a window" → **Programming** → **Groups**. Groups are shown as **tiles**, sorted by number:

- the **name** on two lines, typed directly;
- the **number square**, used as the object for the command keypad;
- "**+ Add a group**", the last tile, creates a new group with the current selection.

Recalling a group

A tap on a group **selects its fixtures**, in its order and with its grid (see the Grid). The selection rule applies: as long as no value has changed, the selection adds to the previous one.

A fixture removed from the patch stays in the group: it is ignored on recall and comes back if it is repatched.

The bottom band

A band at the bottom of the tile shows how much of the group the **current selection** covers, however it was made (the group, fixture by fixture, a range from the keypad...):

Band	Meaning
sand	some of the group's fixtures are selected
blue-gray	all of them, but in a different order
green-gray	all of them, in the group's order
none	no fixture of the group is selected

Extra selected fixtures do not count.

Storing a group

1. Select the fixtures, in the order you want.
2. **Store**, then a tap on the group's tile (or type its number).
3. If the group already exists, choose:
    - **Replace**: it becomes the selection;
    - **Merge**: the missing fixtures are added at the end (their grid, to the right of its own);
    - **Remove**: the selected fixtures that are already in it are taken out (offered only if there are any). The cross cancels.

With no fixture selected, nothing is stored: a message says so in the command line.

"**+ Add a group**" does the same on a free number, without asking.

Delete, copy, move

Delete, **Copy** and **Move** follow the procedure common to all lists: see The lists. Everything goes through the history (**Undo** / **Redo**).

Elsewhere in the software

- **Off** then a group: its fixtures leave the programmer and the selection.
- A group can carry an intensity **Master** on a Fader or a Button: see Masters and Speed Masters.
- **Assign**, a group, then a World: the World takes the group's fixtures. See Worlds.
- Group tiles can also be placed in a **Layout**: see Layout.
- An **Action** can select, add, store or name a group: see Actions.
- The **recipes** of a cue remember the groups its values come from: see Recipes.

Groups are saved in the show.

Presets

A preset keeps **values** (and **effects**) of an attribute family, fixture by fixture: a color, a position, a gobo... You recall it with a tap, and the cues that use it stay readable. As on grandMA, there is **one preset pool per family**.

The preset windows



"Add a window" → **Presets** offers one window per family of the patched fixtures: **Dimmer**, **Position**, **Color**, **Gobo**, **Beam**... (families specific to some profiles included). Each window is titled with just its family name.

Presets are shown as **tiles**, sorted by number: the **name** (typed directly) and the **number square**. "**+ Add a preset**" creates one with what is active in the programmer.

Families and attributes are referred to by their GDTF key, not by a number specific to the machine: a show stays usable on another computer.

Storing a preset

1. Set the values (and effects) of the family in the programmer.
2. **Store**, then a tap on the preset (or its number).
3. If the preset already exists: **Replace**, **Merge** (the new values are added to its own or replace them), or **Remove** (the selected values are taken out, offered only if one of them is in it). The cross cancels.

Only the **active** values of the **pool's family** are stored: a color preset never takes the position. With nothing active in the family, nothing is stored (message in the command line).

The preset also keeps **each fixture's cell** in the selection grid.

Recalling a preset

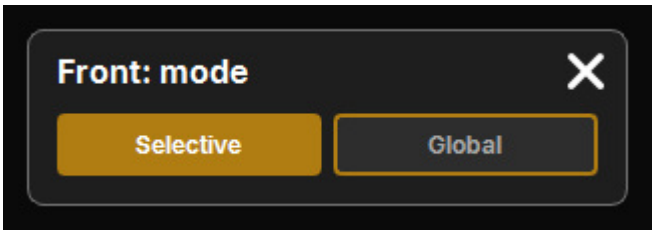
A tap on the preset's square gives its values and effects to the **selected fixtures**. What they receive depends on its **mode**:

Mode	Recall
Selective (default)	each selected fixture **that is in the preset** gets its own value back. With no selection, the preset's fixtures are selected first, with its grid

Global ("G" on the square)	any selected fixture **that has the attribute** receives the value of the first stored fixture: in its function if it has the same profile and mode, otherwise converted, limited to its range

A **long press** on the preset opens the mode choice. Values are always kept per fixture: changing mode loses nothing.

Recalled values keep their **link to the preset**: the Programmer tile shows the preset's name, and if the preset changes, the programmer values that come from it follow.



Two presets on the same attributes

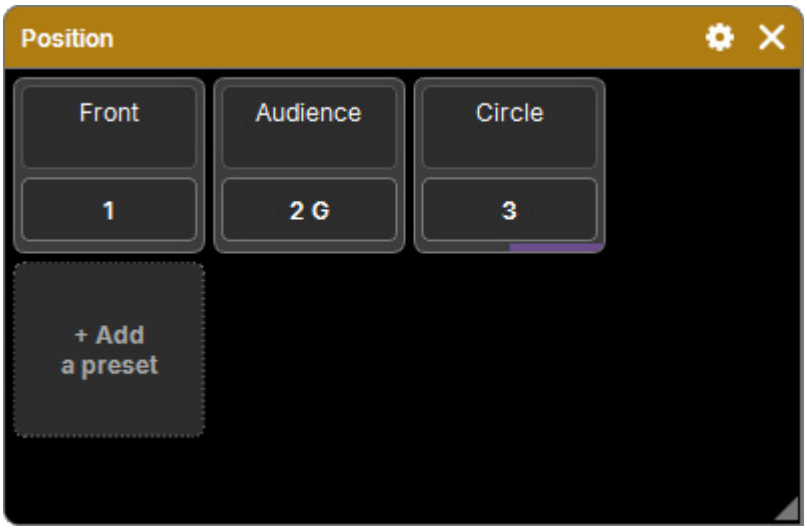
In the programmer, a cell (fixture, attribute) has only one value: recalling a preset **replaces** the previous one's for the fixtures it receives. A color preset recalled after another one therefore takes its place on all the fixtures it covers; the fixtures it does not cover keep their previous value.

- Selection empty between the two recalls: the second preset selects its fixtures and sets them all.
- Selection kept: only the selected fixtures change.

Removing a preset from the programmer

Off then a preset: everything that comes from it (values and effects) leaves the programmer.

What the tiles show



- **Color pool**: the tile background shows the colors the preset gives, in bands proportional to the number of fixtures of each color (Global: the color of the first fixture). The color is calculated from each fixture's real mix (emitters and

filters read from the GDTF, color wheel).

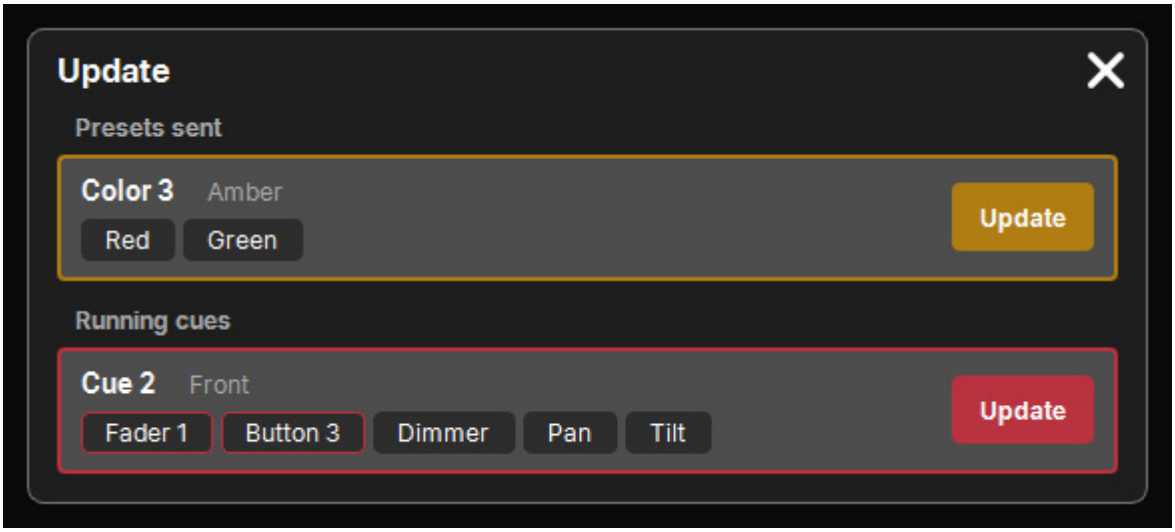
- **Preset sent**: as long as the programmer keeps active values that come from it, a **band** at the bottom of its tile, in the programming color. Clear (values deactivated or emptied) removes it.
- **Preset with effects**: its band is always there, **dull purple**, and **bright purple** when its effects are sent.
- **Values and effects together**: the band is in two parts, values on the left, effects on the right.

Presets and effects

A preset also keeps the active **effects** of its family, in full (steps, speed, Pattern...). Recalled, the effect comes back to the selected fixtures; in Global mode, the effect of the first fixture applies to all. See Effects.

In the Programmer's **Effect mode**, a tapped preset is not sent: it becomes **the current step** of the selection's effects ("‹ Step 2 ›"). The effect then follows the preset if it changes.

Updating: Update



Update (grandMA's Update) updates what is **in play** with what you have just retouched in the programmer. A popup of cards offers:

1. the **presets sent** that active values or effects come from (a value retouched by hand also counts if the preset contains its cell);
2. then the **current cue** of each running Cue List.

Each card shows its title ("Color 3 · Red"), the attributes that change, and **Update**, which does the store (one history entry) and closes the popup. Only the cells the target already has are updated.

Delete, copy, move

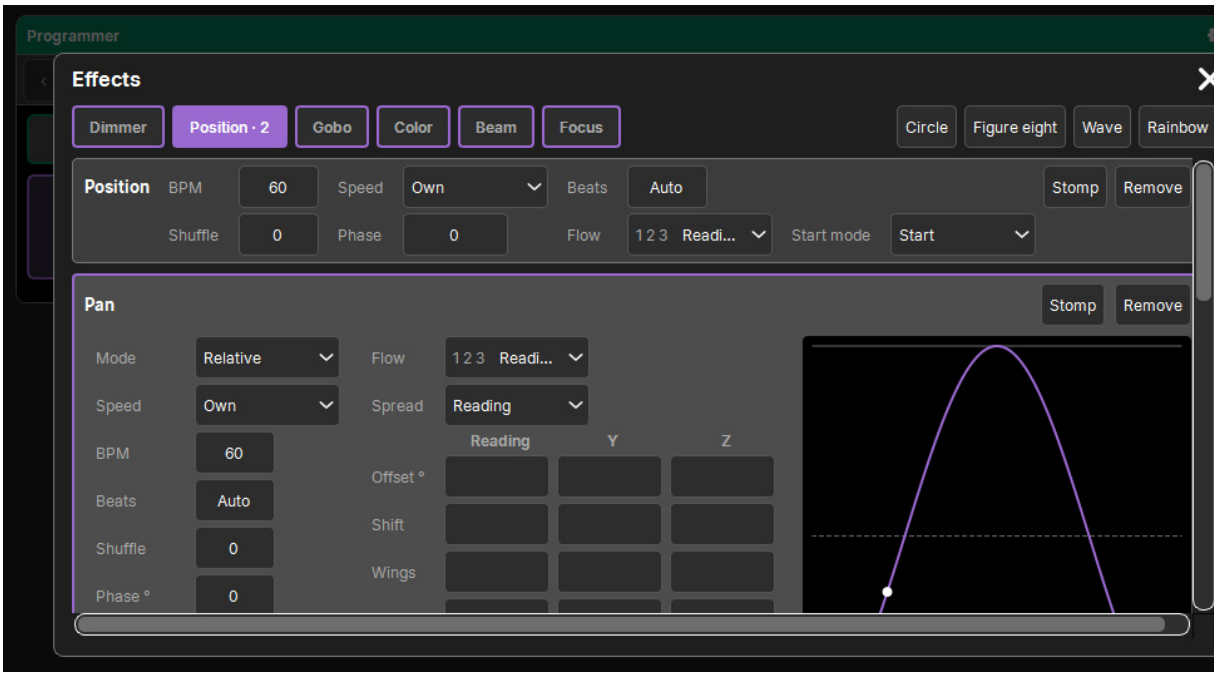
Delete, **Copy** and **Move** follow the common procedure: see The lists. The presets of a family that leaves the patch stay in the show.

Preset tiles can also be placed in a **Layout** (see Layout), and an **Action** can recall, turn off, store or name a preset (see Actions).

Effects

E&S Lighting effects combine grandMA2 effects and grandMA3 phasers. An effect is **one line per attribute** (Dimmer, Pan, Tilt, Red...) made of **steps** that loop, with its speed, and offset from one fixture to the next by the Pattern.

Opening the effects

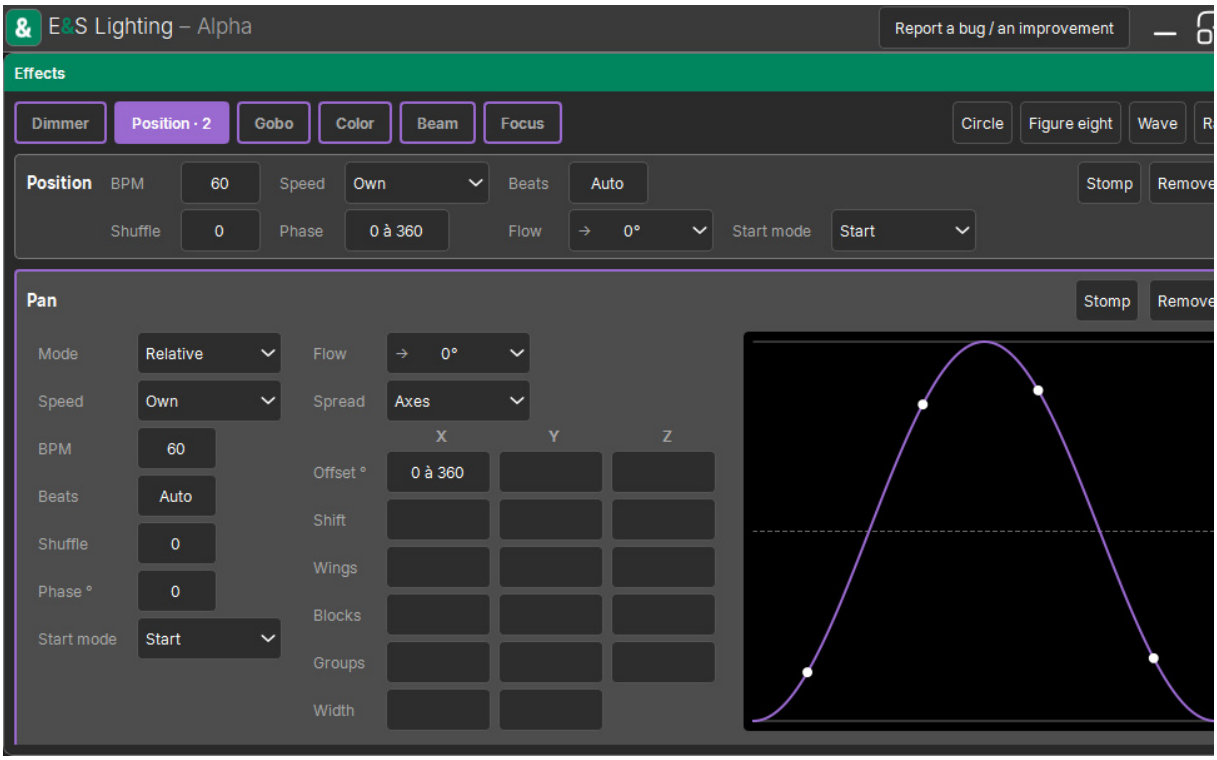


Two paths lead to the same **editor**:

- in the **Programmer** window, the **Effect** button switches the tiles to **Effect mode** (dark purple background); a tap on an attribute's **name** opens the large effects popup, on that attribute. It follows the programmer and closes if the selection changes;
- the **Effects** window (Programming), to place in a View, shows the same editor permanently and follows the selection.

In Effect mode, the Programmer tiles show what the **current step** contains (its value or the name of its preset; "–" if there is nothing), and are outlined in purple if the attribute has an effect (darker if it is inactive). The toolbar shows "**‹ Step 1 ›**" and the **transition** of that step (see The steps).

The editor



- **At the top**: the **families** of the selection as categories ("Position · 2": two lines have an effect there) and, on the right, the **ready-made shapes**.
- **Below**: the **band** of the chosen family, which sets all its lines at once.
- **Then the lines** of the family: those that have an effect, the tapped attribute, and "**+ Shutter**"... to add its other attributes.

A line applies to **the whole selection**. The editor, like the programmer, is outside the history.

Ready-made shapes

Shape	What it gives
Circle	Pan and Tilt as sines, Tilt offset by 90° (relative, ±30° or ±15% depending on the range)
Figure eight	a figure eight in Pan / Tilt, Tilt twice as fast
Wave	a wave offset from 0 to 360 over the selection
Rainbow	three steps red, green, blue (or cyan, magenta, yellow depending on the fixtures)

A shape gives the lines of the attributes the selection has; the others are ignored.

An effect line

Each line shows its name, **Stomp**, **Start** / **Remove**, then:

- on the left its settings;
- in the middle its copy of the **Pattern** (the same table as the Pattern panel);
- on the right the **curve** of one cycle, with **one dot per fixture** moving at the pace of what is really output;

- below its **steps**.

With no effect, the line shows the one it would have (relative for Pan and Tilt, bottom to top otherwise): **Start**, or the slightest change, applies it.

Settings

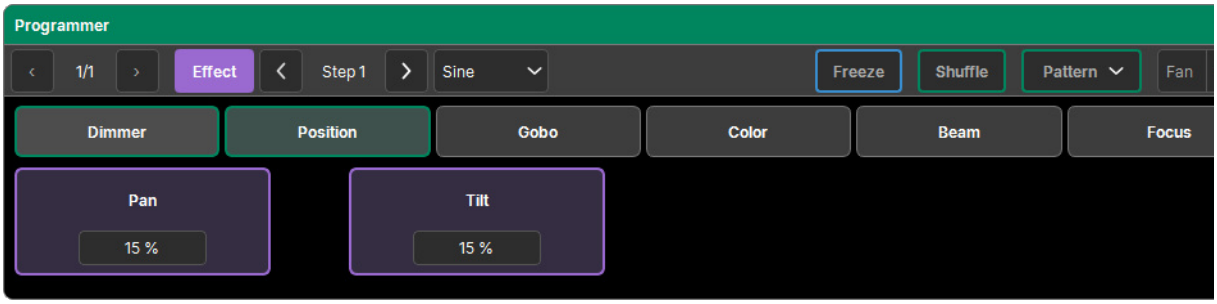
Setting	Role
Mode	**Absolute**: the effect's values replace the fixed value. **Relative**: deviations around what would be output without it (the fixed value, or what is underneath)
Speed	**Own** (the line's BPM) or a **Speed Master**: the line then follows its speed
BPM	beats per minute, 60 by default
Beats	**Auto**: the loop lasts the sum of the Sizes; a number: the loop lasts that many beats, its steps stretched or squeezed
Shuffle	shuffles the fixtures for this line only (a Dimmer that twinkles while Pan and Tilt keep a clean wave)
Phase °	the line's own offset, for all its fixtures (the circle: Tilt at 90°)
Start mode	where the effect starts (see below)

Start mode

Start mode	Behavior
Start (default)	at step 1 when the effect is started: on its cue's Go or when it is given in the programmer; retouching it does not restart it
Synced	on the common clock (or the beats of its Speed Master), as if it had been running in the background: all lines at the same speed stay together
Resume	where it had stopped on this cell
Step N	at the start of that step

Changing the speed of a running effect does not make it jump.

The steps



Steps are counted from 1 and form a **loop**, like a grandMA3 phaser. Each step starts from the previous step (the first one, from the last), reaches its value according to its **transition**, and stays there until the next step.

Column	Role

Transition	**Square** (the value from the start of the step), **Ramp** (even fade over the whole step), **Sine** (smooth fade)
Rise %, **Fall %**	for a Square: soften its edges, in % of the step
Size %	the step's duration in % of a beat: 100% = one beat
Value	in the attribute's unit, or a **preset** of the family (specific to each fixture)

- "**+ Step**" adds a copy of the last one; a step's cross removes it.
- The basic effect: **two Square steps**, 100 then 0, one beat each.
- A true sine: Step 1: 100 as Sine, Step 2: 0 as Sine.
- A single step: a fixed value.
- Between two values of different functions, or on a stepped function (wheel, gobo), a Ramp or a Sine switches halfway.

Filling the steps from the Programmer

In Effect mode, "**‹ Step N ›**" chooses the step your settings go to:

- a **typed value** in a tile goes into that step (a range "0 to 100": each fixture its own);
- a **tapped preset** becomes that step for all its attributes (a color preset: Red, Green, Blue...); a fixture missing from the preset keeps the previous step's value;
- the arrows go up to the last step + 1, to add one;
- the **transition** list next to it follows the step, and also applies to the steps to come.

The family band

It sets **all the lines** of the family at once: **BPM**, **Speed**, **Beats**, **Shuffle**, **Phase** (the Pattern offset on X, "0 to 360"), the Pattern's **Flow**, **Start mode**, **Stomp** and **Remove**. An empty field means the lines have different values.

Stomp

A line's **Stomp** button turns it into an **effect stop** on those cells:

- in the programmer, it also stops the effects that the Cue Lists are running on those cells (their fixed values stay);
- it is a toggle: a second tap restarts the effect as it was;
- it can be given **even without an effect** (on the line, or on the whole family through the band): a **Stomp alone**, which is stored in a cue like an effect. For example on a **Flash** or **Temp** Button, to cut the effects for the duration of the press.

Storing an effect

An effect is stored like a value, with **Store**:

- in a **cue**: see Storing cues;
- in a **preset** of its family: see Presets.

It keeps **everything**: its steps, its speed, its offset, its copy of the Pattern, each

fixture's cell in the grid and the preset it comes from.

Effects in Cue Lists

- Effects have their **own tracking**, separate from that of values: an effect continues from cue to cue until a cue gives another effect to the same attribute, or a **Stomp**. A fixed value does not stop it (a relative effect runs around it).
- **One effect per cell** (fixture, attribute): two effects never stack. Between Cue Lists, the one with the highest priority wins, then the most recent; the others stop, their fixed values stay.
- A cue's effects come in with the cue's fade.

Taking over an effect played by a cue

For an attribute with no effect in the programmer, the editor shows the one a Cue List is running: "**Played by Cue List 1 · Cue 2**". The slightest change puts it in the programmer as it is running, to adjust it; **Update** then puts it back into the cue.

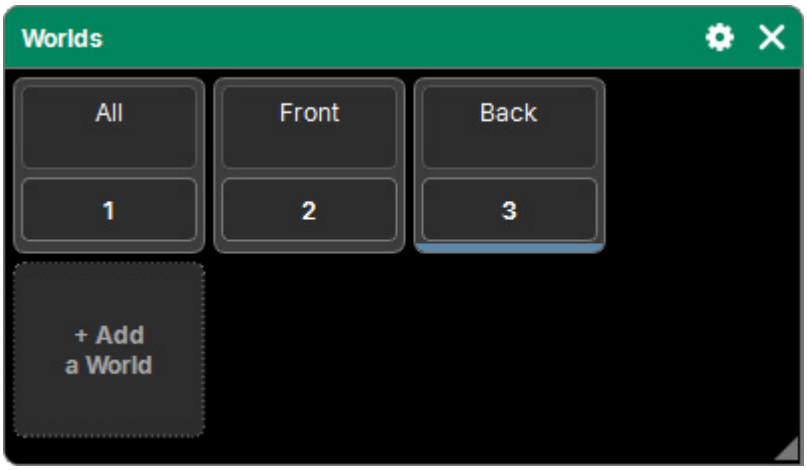
Speed Masters

A line whose **Speed** is a Speed Master follows its tempo: several effects stay in sync, and a tap in rhythm (**Learn**) realigns them all. See Masters and Speed Masters.

Worlds

A World limits what the programmer can touch to **a set of fixtures**, as on grandMA. Useful when several people program different parts of a big show, or to avoid touching the front by mistake while adjusting the backlight.

The Worlds window



"Add a window" → **Programming** → **Worlds**. Worlds are shown as tiles, like groups. The **active** World is marked by its band.

- **1 · All**: all fixtures. It is active at the start, and can be neither modified nor deleted.
- A tap on a tile **makes that World active**.

Creating a World

- "**+ Add a World**": a new World with the current **selection**.
- **Store** then a World: the selection goes into it, **Replace** or **Merge** (the cross cancels). All is refused.
- **Assign**, a **group**, then a World (or "+ Add a World"): the World takes the group's fixtures.

Delete, **Copy** and **Move** follow the common procedure (see The lists). All can be copied (its fixtures at that moment) but not deleted. Deleting the active World makes All active.

What a World limits

A World only affects **the Programmer and the Layouts**:

- a fixture outside the World **cannot be selected**;
- what it had in the programmer (values, effects, place in the selection) is **set aside**: nothing is output from it any more, and nothing from it goes into a store. Everything comes back as soon as a World contains it again, unless a **Clear** has emptied the programmer in the meantime;
- in a **Layout**, fixtures outside the World are shown faded (their opacity is set in the Layout's gear).

Playback is not affected: Cue Lists, Faders and Buttons play all fixtures, whatever

the active World.

A sub-fixture belongs to a World along with its fixture.

Good to know

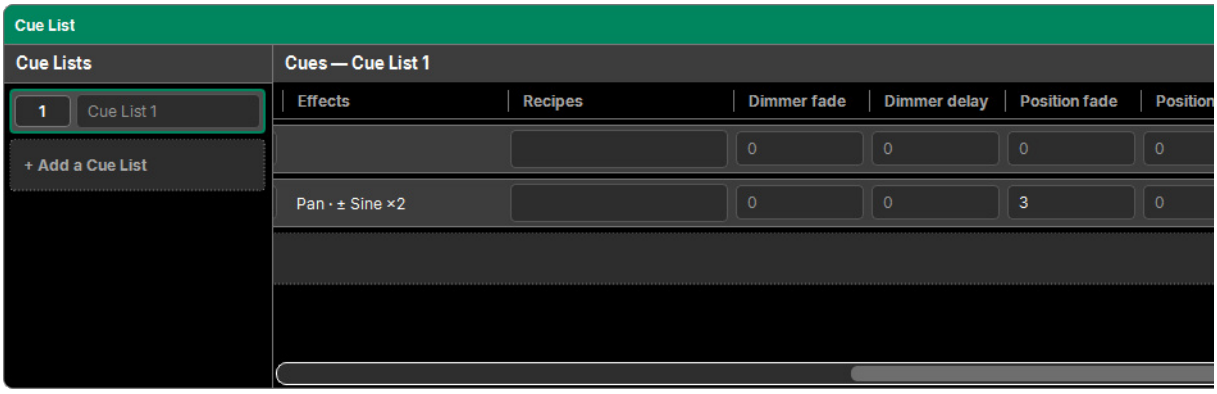
- The active World is a state of the moment, like the programmer: it is not in the history, and when a show is opened, **All** is active.
- The Worlds themselves are saved in the show.
- World tiles can also be placed in a **Layout**; an **Action** can activate a World (see Actions).

Cue List

A **Cue List** is a series of **cues** (the sequence of grandMA): each cue is a lighting state, with its times. You play it from a **Fader** or a **Button** (see Faders and Buttons), an Action or Timecode.

The **Cue List** window is in the **Programming** category of "Add a window".

The window



- **On the left**, the show's Cue Lists. Tapping one shows it on the right and makes it the **active Cue List**: the one the command line targets ("Store Cue 5"...). **+ Add a Cue List** creates a new one.
- **On the right**, the cues of the Cue List shown, as a table. **+ Add a cue**, at the bottom, stores the programmer into a new cue, at the next whole number after the last one (see Storing cues).
- The column widths and the Cue List shown are remembered with the window (and in Views). Several Cue List windows can show different Cue Lists.

A Cue List without a name shows as "Cue List 1", a cue without a name as "Cue 1.5": just type a name in its cell.

Cue numbers

- A number goes from 0.01 to 9999.99, with at most two decimals: `1`, `1.5`, `2.25`.
- A decimal cue ("1.5") is a cue in its own right, slipped in between 1 and 2.
- Cues are always sorted by number.

The columns

Column	What it holds
No.	the number: it is the cue's button (see below)
Name	its name, typed directly
Col.	the color of its card (a tap opens the color picker)
Trig.	its trigger: **Go**, **Time** or **Follow**
Mode	**Normal**, **Assert** or **Break**
Wait	the wait time of a Time or Follow trigger

Fade	the in fade
Delay	the delay before the in fade
Out fade	the fade of intensities going down (empty: same as the fade)
Out delay	the delay of intensities going down (empty: same as the delay)
Duration	the total length of the transition (calculated, read only)
Note	free text
Effects	its effects, summarized (read only)
Recipes	its recipe lines, summarized; a tap shows them all (see Recipes)

Then, for each attribute family present in the Cue List (Dimmer, Position, Color...), two columns: its own **fade** and **delay** (see below).

Times

- Times are typed in seconds (`2.5` or `2,5`) or in minutes:seconds (`1:30`), to the hundredth of a second, up to 9999.99 s.
- A cleared **Fade**, **Delay** or **Wait** is 0.
- A cleared **Out fade** or **Out delay** takes the in time.
- The **Duration** is the longest delay + fade of the cue: in, out, family or value times.

The trigger

Choice	The cue starts...
Go	by hand: a Go (a Fader's button, an Action, Timecode)
Time	on its own, its **Wait** after the **start** of the previous cue
Follow	on its own, its **Wait** after the **end** of the previous cue (its duration)

A chain of Time or Follow cues runs by itself after a Go. Nothing starts during a Pause or a manual X-Fade; the remaining time resumes afterwards.

The mode

Mode	Effect
Normal	the cue only keeps what changes (tracking)
Assert	when it starts, all the values of its state (including those it inherits) come back on top: they take over from the programmer and from the other Cue Lists
Break	tracking stops at this cue: it inherits nothing from the cues before, and the following ones track from it

Tracking: as on grandMA, a cue only keeps the values that change compared with the previous cues. A value stays on stage until a cue changes it. The Cue List's **Tracking** setting (below) turns it off.

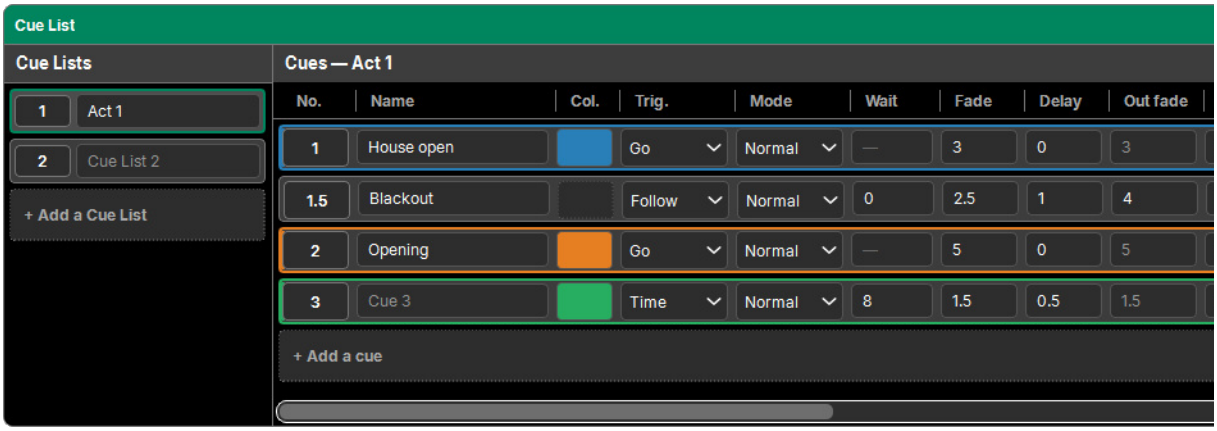
Times per family

A family's fade and delay columns give all the values of that family their own times in the cue: for example a **Position fade** of 5 s in a 2 s cue. Cleared, the cell takes

the cue's times.

A value can also have its own time, given by the Pattern when it is stored: the cue's fade or delay cell then shows it on hover ("Pattern fade, per fixture"). Order of priority: the value's time, then its family's, then the cue's.

Gestures on a cue



- **Tapping the number** of a cue (with no pad key): the Cue List **goes there directly**, with no delay or fade. Handy for finding your place in rehearsal.
- With a pad key, the number is the object:
  - **Store** then the number: the programmer goes into that cue (see Storing cues);
  - **Off** then the number: the Cue List stops;
  - **Delete**, **Copy**, **Move**: see Storing cues.
- **Cells** (name, times, note) are edited in place. With several cells selected (a rectangle drawn with a finger or the mouse, or a column title for the whole column), the typed value goes **to all of them**, in a single history step. The same choice (trigger, mode, color) applies in the same way to the whole selection. See Selecting.

On a Cue List in the left column:

- with no key, a tap shows it;
- **Off** then the Cue List: it stops;
- **Assign** then the Cue List: it becomes the source, to be placed afterwards on a Fader or a Button;
- **Delete**, **Copy**, **Move**: the common list sequence (see Delete, copy, move).

The current cue and the next one

A Cue List in playback has a **current cue**, the same everywhere (Cue List window, Faders, Buttons): it is the Cue List that plays, not the Fader.

- The current cue is highlighted in the table. When it starts, a colored band moves from right to left over its whole duration (delays included): you can see the cue coming in.
- The **next cue** is the one the next Go will start: the next one in order, or the first one after the last if the Cue List loops.

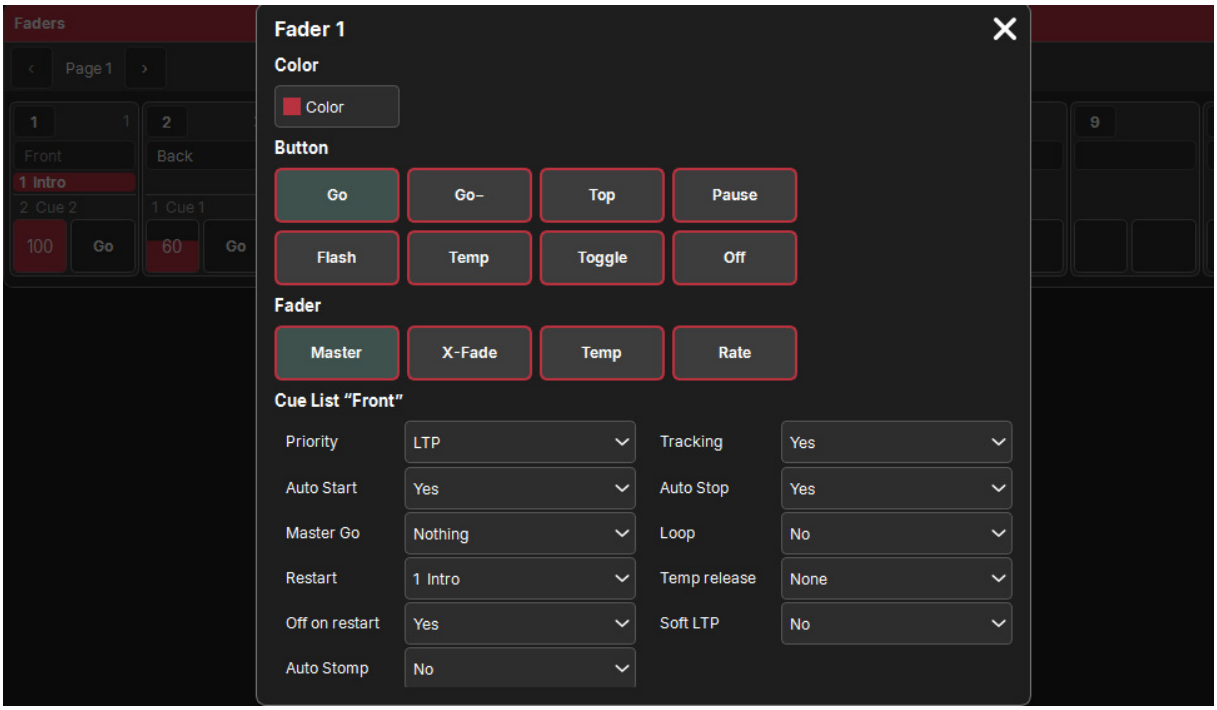
Playing a Cue List

The playback commands are on the buttons of Faders and Buttons, and in Actions:

Command	Effect
Go	the next cue, with its times. Stopped: it starts again according to its **Restart**. Paused: it resumes
Go–	the previous cue, with its times
Top	the first cue, with its times
Pause	freezes the running fades; Go or Pause starts them again from where they were
Off	stops the Cue List at once

The others (Flash, Temp, Toggle, and the fader functions) are described in Faders and Buttons.

Cue List settings



They are set in the settings popup of a Fader or Button that holds the Cue List (a tap on its number), section **Cue List "..."**:

Setting	Default	Effect
Priority	LTP	its output priority: Very low, Low, LTP, High, Very high, HTP, Swap, Super
Tracking	Yes	No: each cue only gives its own values, nothing from the cues before
Auto Start	Yes	a fader leaving 0 starts the stopped Cue List
Auto Stop	Yes	a fader brought back to 0 stops it

Master Go	Nothing	without Auto Stop, what a fader leaving 0 does on a Cue List that is playing: Nothing, Go, On (reloads the current cue), Top
Loop	No	a Go on the last cue goes back to the first
Restart	the first cue	where a stopped Cue List starts again: **Current cue** (where it was), **Next cue**, or a chosen cue
Temp release	None	where the Cue List goes when a Temp is released: None (it fades out) or a chosen cue
Off on restart	Yes	when all its cells are taken over by other Cue Lists, it goes off
Soft LTP	No	on the fader, its attributes other than intensity go from what is underneath to its value, following the fader's travel (otherwise: as soon as the fader leaves 0)
Auto Stomp	No	its fixed values stop the effects of other Cue Lists on the same cells (otherwise: a fixed value goes under an effect of another Cue List of the same or lower priority)

Priorities: cell by cell, the highest priority wins; at equal priority, the most recent value (LTP). In **HTP**, intensities take the highest value. **Swap** puts LTP intensities above HTP. **Super** goes above everything, programmer included. The programmer has no priority: against the Cue Lists, the most recent value wins (except in **Freeze**, see Programmer).

Set for several selected Faders or Buttons, the settings apply to all their Cue Lists (except Restart and Temp release, which name a cue).

A damaged cue

If the show file has been damaged (power cut while writing, full disk...), unreadable cues are reported when the show opens (**Damaged cues** popup): what is missing stays empty, the rest of the show is untouched. Storing the cue again replaces what is damaged.

Storing cues

Storing means putting into a cue what is **active in the programmer**: the values and effects you have given to the fixtures (see Programmer). Values keep the preset they come from: the cue refers to it, and a preset changed later changes the cue.

As on grandMA, storing works in **tracking**: only the values that change the state of the Cue List just before the cue go into it. What is already on stage from the previous cues is not copied.

A new cue

Several ways, as you like:

Gesture	Where the cue goes
+ Add a cue (at the bottom of the cue table)	at the whole number following the last cue
Store Cue `5` **Enter**	cue 5 of the active Cue List
Store Cue `1.5` **Enter**	a decimal cue, between 1 and 2
Store then the button of a **Fader** or a **Button**	a new cue at the end, in its Cue List (see below)

The **active Cue List** is the last one shown in a Cue List window (a tap on it in the left column). Without an active Cue List, the command line says "No Cue List shown".

A new cue is created with what is active, even if nothing is (an empty cue).

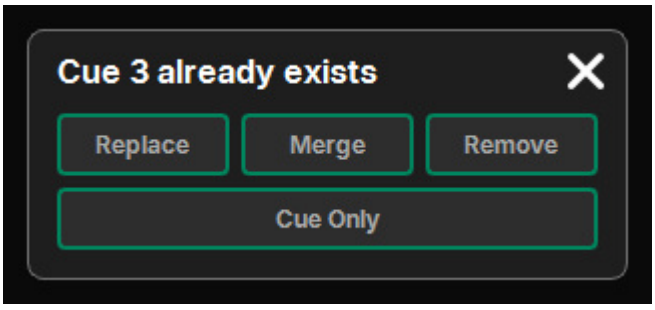
Between two cues

A new cue slipped **before** an existing cue (for example 1.5 between 1 and 2) asks a question:

- **Normal**: the new cue plays in tracking, cue 2 and the following ones inherit what it changes;
- **Cue Only**: the new cue changes nothing for what follows. Cue 2 receives the old state of everything 1.5 changed.

This question is not asked if the Cue List is not in tracking, or if the next cue is in **Break**.

Storing into an existing cue



Store then the number of a cue that already has values: a popup asks what to do.

Choice	Effect

Replace	its values and effects become those of the programmer
Merge	those of the programmer are added to it (same cell: the new value wins)
Remove	the same (fixture, attribute) pairs as in the programmer **leave** the cue; offered only if the cue has some

The **Cue Only** switch, below these choices, protects what follows: the next cue receives the old state of what changes.

- With nothing active in the programmer, storing is refused ("no active value").
- A cue that is still empty is filled without a question.
- The popup's cross cancels.

A range of cues

Store Cue 3 **Thru** 7 **Enter**: the programmer goes into **all existing cues** from 3 to 7, each one compared with the state just before it. If one already has values, a single question applies to all (Replace, Merge, Remove, Cue Only). The whole thing is a single history step.

Only Store accepts a range of cues.

Storing on a Fader or a Button

Store then the **button** of a Fader or a Button:

- it holds a Cue List: a new cue is created at the end of that Cue List;
- it is **free**: a new Cue List is created, the programmer goes into its cue 1, and the Fader (or Button) receives it. All in one history step;
- it holds a Master or a Speed Master: refused ("it is not a Cue List").

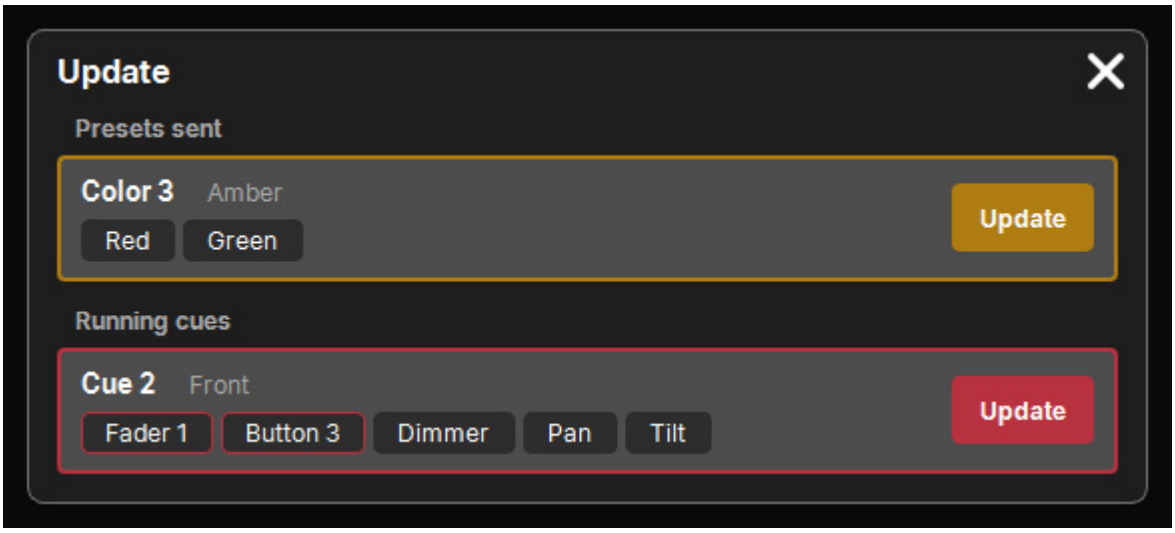
The values' own times

If the programmer's Pattern gives times per fixture (spread fades or delays), each value keeps its time in the cue. It comes before the family's times and the cue's times (see Cue List).

Recipes

Along with the values, the cue keeps **how** they were made: "Group 1 → preset Red". If the group changes later, the cue can be redone (rebaked) for its new fixtures. See Recipes.

Update



The pad's **Update** key updates what is **in play** with what is active in the programmer, like Update on grandMA. A popup lists the possible targets, one card each:

1. first the **presets** sent to the programmer (an active value comes from them, even if changed since);
2. then the **current cue** of each Cue List that is playing.

Each card says what the target is ("Color 3", "Cue 2" and its Cue List), the Faders or Buttons that hold it, and the attributes that change. **Update** applies it and closes the popup.

- Only the cells the target **already** has are updated: Update does not add new fixtures or attributes.
- In a cue, it is a **Merge** in tracking.
- A target with nothing to update is not offered; with no target at all, a message says so.

Delete, copy, move

Cues follow the common list sequence (see Delete, copy, move):

- **Delete** then a cue (or selected cues, then **Enter**): they are deleted. On the pad: **Delete Cue** `4` **Enter**.
- **Copy** or **Move** then a cue (the source), then **Cue** `N` **Enter**: the cues arrive at N, keeping their spacing. A copied Cue List gets "(copy)" after its name.
- If cues already occupy the destination, a popup asks:
  - **Replace**: they are overwritten;
  - **Merge**: the destination cue keeps its times, color, note and trigger, receives the incoming values, and its name becomes "old name + incoming name";
  - **Append**: the cues arrive after the existing ones.

All of this is a single history step: **Undo** undoes the whole operation.

A cue's number is its place in the Cue List: moving a cue means renumbering it. Timecodes and Actions that refer to it follow its new number.

Recipes

A cue keeps its values, but also **how** they were made: these are its **recipes** (the Recipes of grandMA3, made automatically here). When a group changes (a fixture added, removed, moved in its grid), the cues that use it can be **rebaked**: their values are remade for today's fixtures, without storing everything again.

What a recipe remembers

Each gesture made in the programmer on a selection that came from a **group** (or from the fixtures of a preset recalled without a selection) makes a **recipe line**:

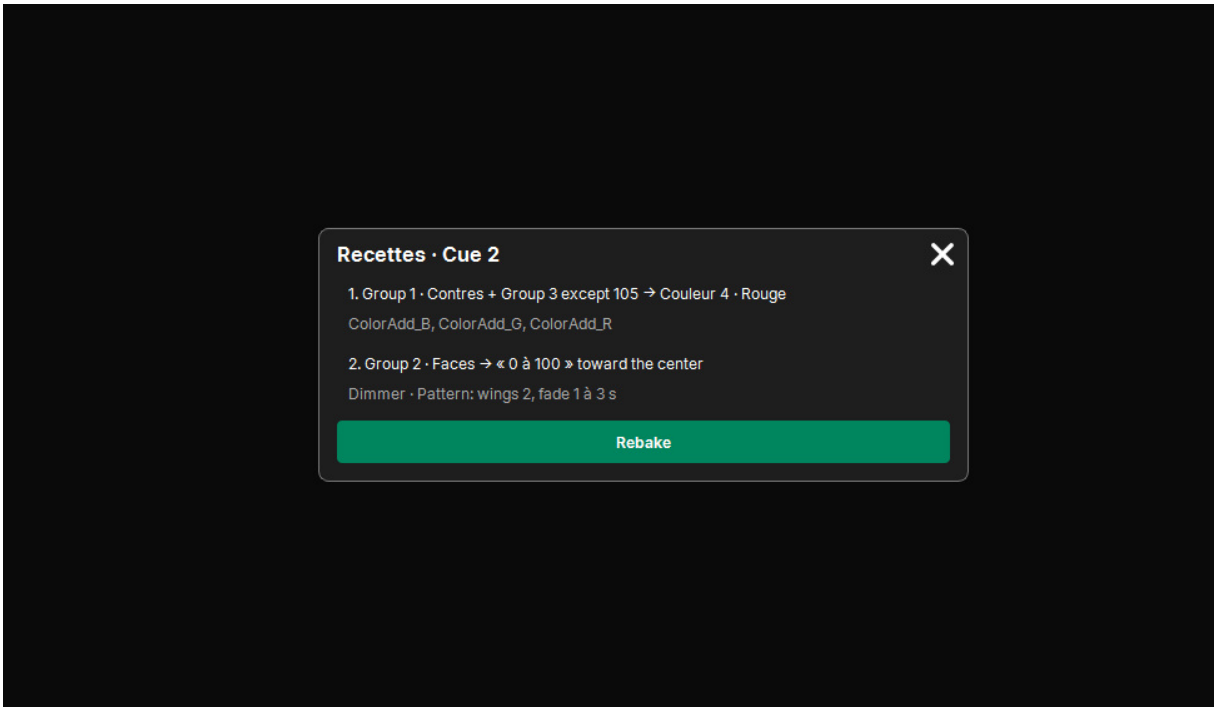
- "Group 1 → Color 4 · Red": a recalled preset;
- "Group 2 → Dimmer 0 thru 100": a typed value, spread according to the Fan and the Pattern;
- "Group 3 → Open": a ready-made value from the profile;
- "Group 1 + Group 3 except 105 → ...": a selection of several groups, minus a fixture removed by hand;
- an effect given or adjusted (its effect line is kept).

The line also remembers how the selection was arranged in the selection grid (see Programmer; the order of the groups, the shuffle, a grid adjusted by hand) and the Pattern. One line per gesture, never per fixture: a show stays light.

A selection made **by hand** (fixture by fixture, on the pad, in a Layout) makes no recipe: its values stay as stored.

The programmer's recipe log starts again from zero at each **Clear**.

In the Cue List



When storing, the cue keeps the recipe lines its values come from (see Storing

cues):

- **Replace**: the programmer's lines replace its own;
- **Merge** and **Update**: the programmer's lines are added at the end; an identical line is not doubled;
- **Remove**: what leaves the cue also leaves its recipe (a rebake will not put it back).

The **Recipes** column of the cue table summarizes them. It turns **orange** when the cue needs rebaking (one of its groups has changed). A tap on it opens the **Recipes · Cue ...** popup: all its lines in order, each in one sentence, then the details in a lighter style (its attributes, its Pattern, the adjusted or shuffled grid, the fixtures removed from the cue). The **Rebake** button redoes it at once.

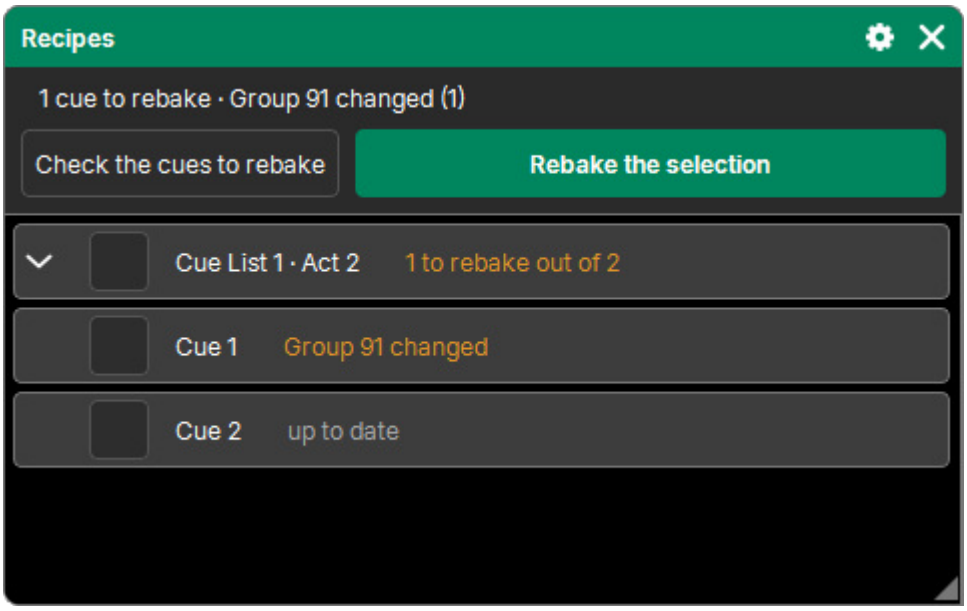
Rebaking a cue

Rebaking replays each recipe line of the cue in a working programmer, with today's groups and presets, then puts the result into the cue. The rules, as on grandMA:

- a value made **by hand** on the same cell **wins**: the rebake does not touch it;
- a fixture **removed** from the cue (Remove) does not come back;
- a fixture that **left the group** leaves the cue; a fixture **added** to the group receives its value;
- what does not change the state before the cue does not stay in it (tracking);
- an effect already there keeps its content, its place follows the new grid;
- a line whose group has been deleted is ignored: its values stay.

The result is shown in the command line: "3 values added, 1 value removed", "already up to date"... Rebaking is a history step: **Undo** undoes it.

The Recipes window



"Add a window" → **Programming** → **Recipes** goes through the whole show:

- the Cue Lists that have recipes, each followed by its cues (an arrow expands it);

- the state of each cue: **to rebake** (in orange, with the cause: "Group 9 changed"), **to check** (stored before groups were tracked, or group deleted), or **up to date**;
- a checkbox per cue; checking a Cue List checks all its cues;
- **Check the cues to rebake** checks at once all those that need it;
- **Rebake the selection** rebakes them all, with a progress bar; **Cancel** stops it partway. The whole thing is a single history step. A cue changed in the meantime is left as it is (the message says so).

The checked boxes and the expanded Cue Lists are shared by all open Recipes windows.

A recipe is never applied on its own: as long as you do not rebake, the cue stays exactly as stored. The orange column and the Recipes window only tell you it could be remade.

Faders and Buttons

Faders and **Buttons** are the executors of grandMA: numbered slots that play a Cue List, control a group's Master or a Speed Master.

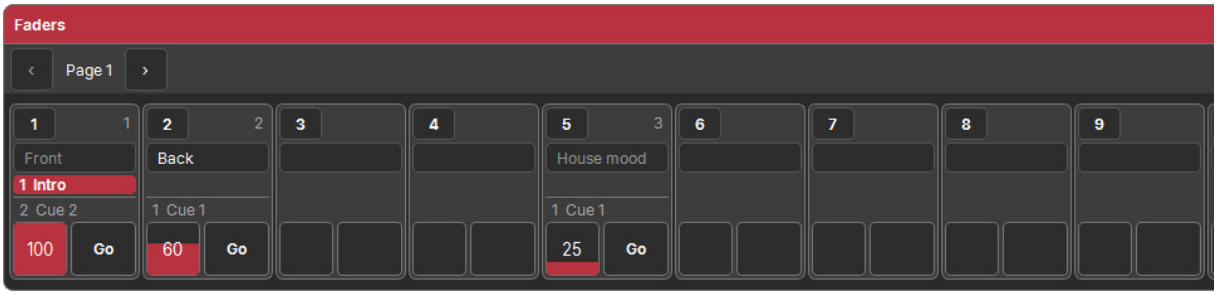
- a **Fader** has a level square and a button;
- a **Button** only has the button.

The **Faders** and **Buttons** windows are in the **Playback** category.

Pages and numbers

- At the top of the window, **‹ Page 1 ›** changes the page. The page is **shared**: changing it changes it in all Faders and Buttons windows, on all screens.
- Each screen shows **18** slots of the page, in sequence: the main screen numbers 1 to 18, the next screen 19 to 36, and so on.
- A window that is too narrow scrolls its slots horizontally, by dragging.
- What a slot holds (its Cue List, its function, its color) is kept in the show. Its level, however, is the current state: it is not saved.

A slot



From top to bottom:

1. on the left, the slot's **number** (a tap: its settings); on the right, the number of its Cue List;
2. its **name**, always editable. Empty, it shows its Cue List's name in gray;
3. the **current cue**, in bold, then the **next cue**, in a lighter style. When a cue comes in, its background moves from right to left over its whole duration. A tap on this area opens the cue table;
4. for a Fader, the **level square**: it fills from the bottom, with the percentage written in the middle. Drag in it to set the level: at the top 100 %, at the bottom 0;
5. the **button**, with its function written on it ("Go", "Flash"...).

A free slot only has its number.

Giving a Cue List to a slot

- **Assign**, then a Cue List (in the left column of a Cue List window), then the **button** of a Fader or a Button: it receives it.
- Or **Store** then the button of a **free** slot: a new Cue List is created with the programmer in its cue 1, and the slot receives it (see Storing cues).

The same Cue List can be on several slots, on several pages: its current cue is the

same everywhere, because it is the Cue List that plays, not the Fader.

The button and the pad

With no pad key, a tap on the button performs **its function** (below). With a key, the button is the object of the action:

Key	Effect on the slot
Assign	receives the source Cue List; with no Cue List as source, it becomes the source itself, to be placed in a Layout
Store	the programmer goes into a new cue of its Cue List
On	its Cue List turns on at once, with no fade, at its start cue
Off	its Cue List stops
Delete	the slot is freed (the Cue List itself stays)
Copy, **Move**	the common sequence: see Delete, copy, move

Button functions



For a Cue List:

Function	When pressed
Go	the next cue, with its times (stopped: it starts according to its Restart; paused: it resumes)
Go–	the previous cue, with its times
Top	the first cue, with its times
Pause	freezes the running fades; a new press (or Go) starts them again
Flash	while pressed, the Cue List is at full and its values come on top, with no times; stopped, it starts at its start cue. On release, everything goes back to how it was (started by the press: it stops)
Temp	while pressed, the Cue List is on, with its times. On release, it goes to its **Temp release** cue if it has one, otherwise it fades out with its cue's out times
Toggle	stopped: it starts (with its times); playing: it fades out with its cue's out

	times
Off	it stops

For a group Master: **Black** (at 0 while pressed), **Flash** (at full while pressed), **Toggle** (cut, then restored).

For a Speed Master: **Learn** (tap the tempo in rhythm), **×2**, **÷2**, **60 BPM**.

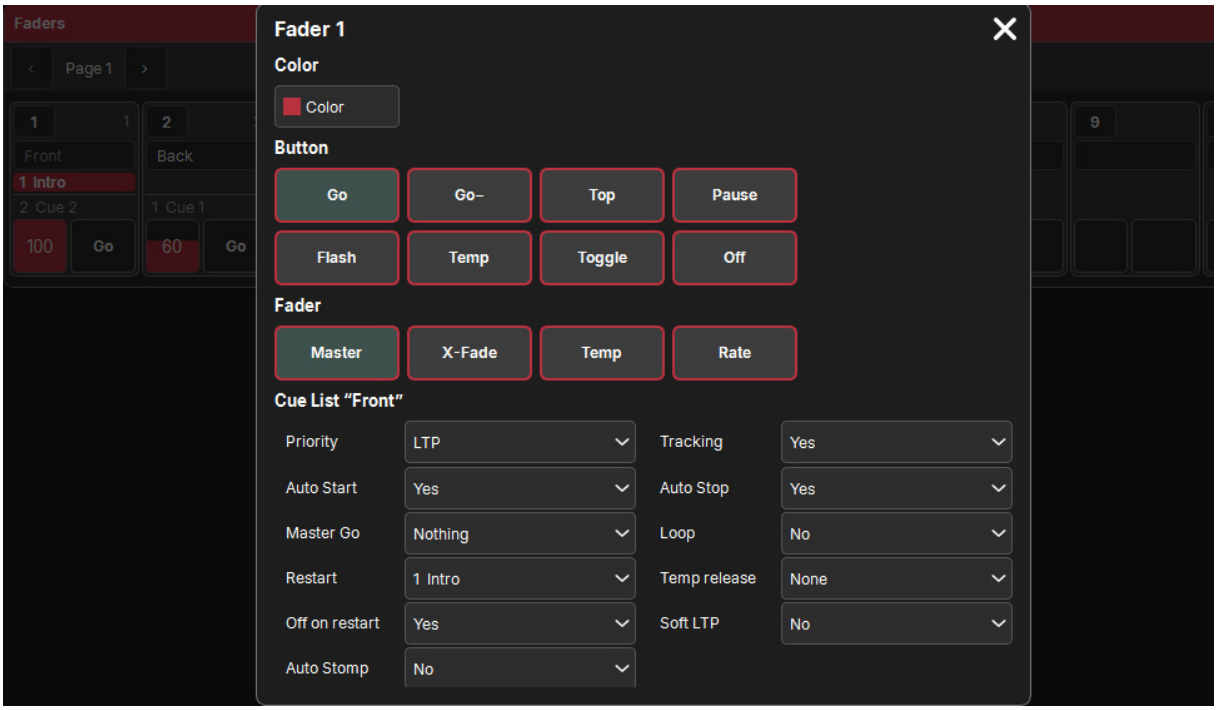
Fader functions

Function	The square…
Master	gives the Cue List's level. Leaving 0, it starts it if it is stopped (**Auto Start**); otherwise, it performs its **Master Go**. Brought back to 0, it stops it (**Auto Stop**)
Temp	gives the level; leaving 0 it starts the Cue List, coming back to 0 it releases it (like releasing a Temp button)
X-Fade	makes the transition to the next cue **by hand**: leaving 0 it begins, at the top the cue is played. You then have to bring it back down to 0 before the next one
Rate	sets the speed of its times: in the middle, normal speed; at the bottom, 4 times slower; at the top, 4 times faster

Auto Start, Auto Stop and Master Go are Cue List settings (see Cue List).

A Cue List's level is that of the highest of its Faders in Master or Temp (all pages); with no Fader, or started from a Button, it is at 100 %.

Slot settings



A tap on a slot's **number** opens its settings. With several slots selected (a rectangle drawn over the window), they apply to all of them.

On a **free** slot, the popup offers to give it:

- **Group master**: the list of groups;
- **Speed Master**: the list of Speed Masters, or **+ New** (a new one per slot).

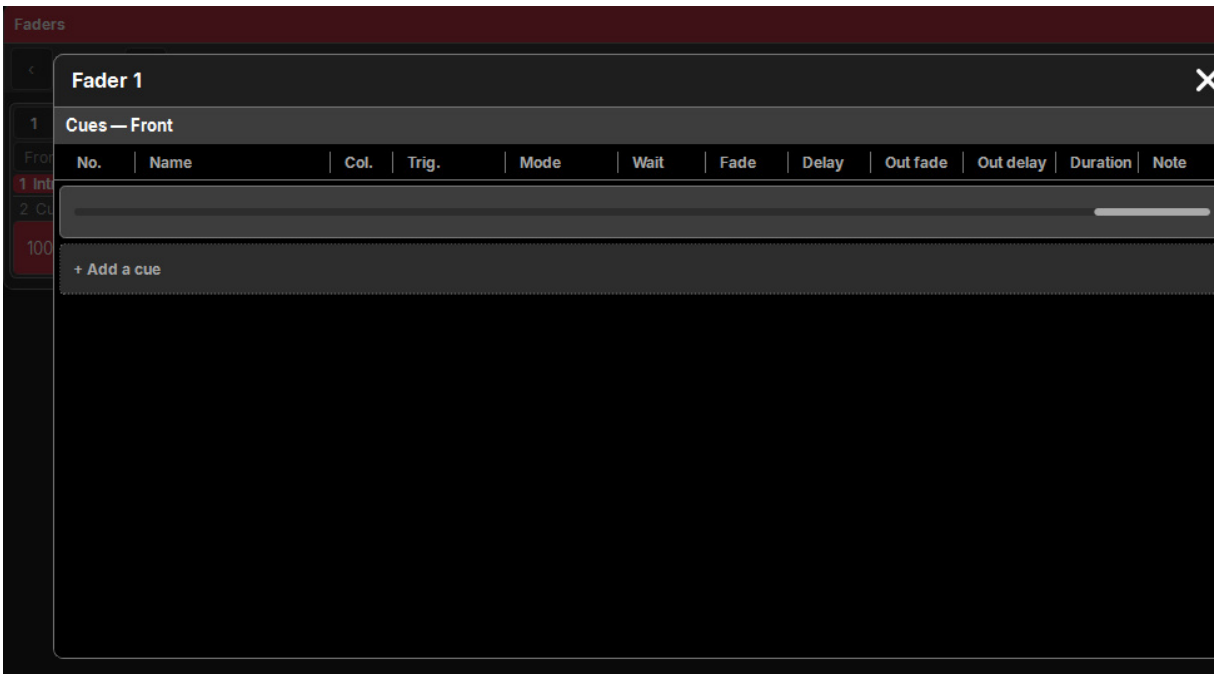
Several free slots selected receive the groups (or the Speed Masters) in sequence, starting from the one chosen.

On an **assigned** slot:

Section	Setting
Color	the color of the slot (by default, that of the Playback category)
Group / **Speed Master**	what a Master or a Speed Master controls
Scale, **Fader range (BPM)**	for a Speed Master: see Masters and Speed Masters
Button	the button's function
Fader	the square's function (Faders only)
Cue List "..."	the settings of its Cue List: see Cue List

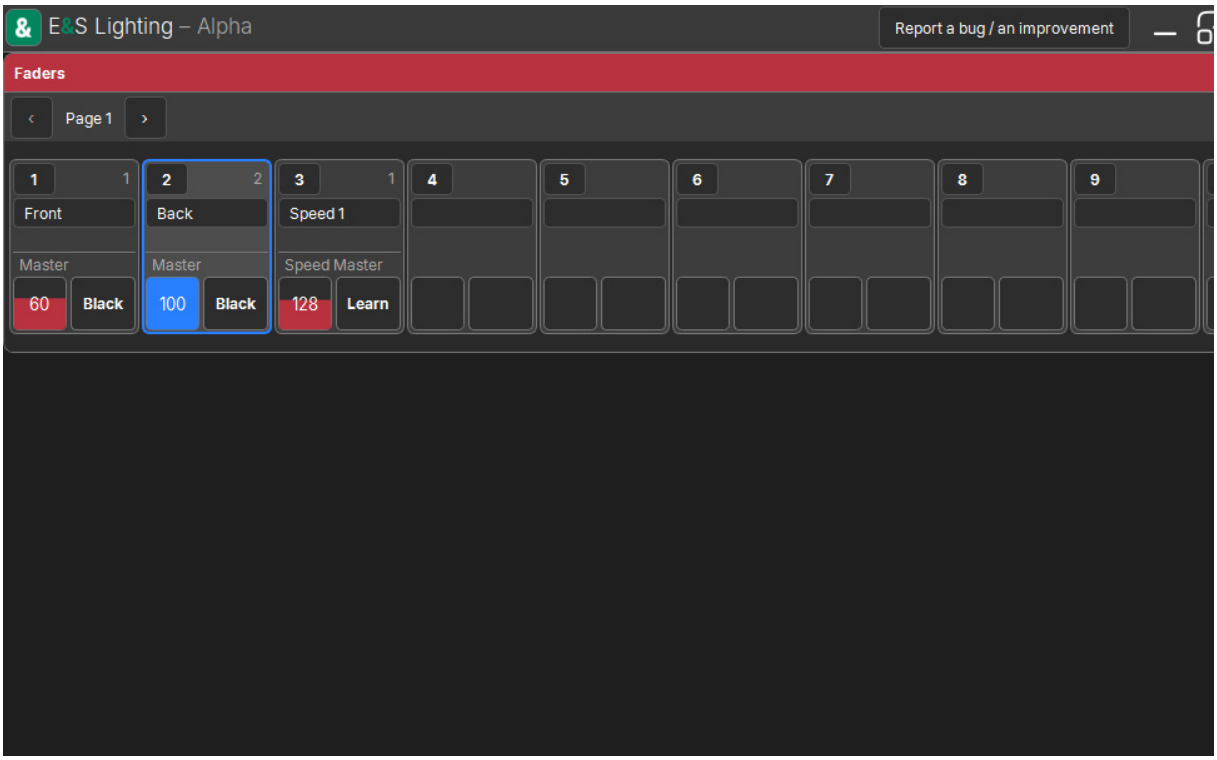
Changing a fader's function puts its square back to its neutral position (in the middle for Rate, at the bottom for X-Fade).

The cue table



A tap on the cue area of a slot opens, in a large popup, the **same table** as in the Cue List window, for its Cue List: same columns, same gestures (go to a cue, Off, edit a cell, Store...). It follows the changes while it is open; the cross closes it.

Masters and Speed Masters



A slot can hold:

- a **group Master**: it multiplies the intensity of the group's fixtures. The Master of the **All** group serves as Grand Master. A Master only counts while a slot holds it: removed, its fixtures go back to 100 %;
- a **Speed Master**: the speed of the effects that follow it.

Their name: a Master shows its group's name as long as you do not give it one; a Speed Master is renamed from its slot. See Masters and Speed Masters.

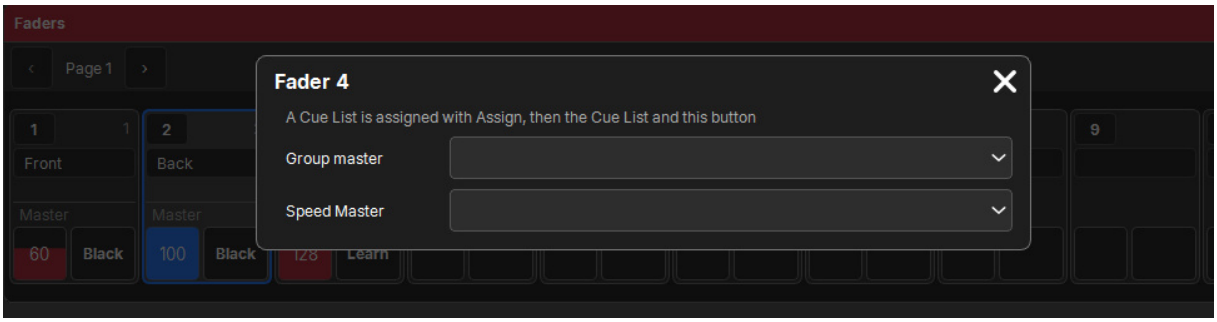
In a Layout

A Fader or a Button placed in a Layout keeps **its page**: it does not follow the page shown. It can take a **Large** appearance there (a large fader with a handle) or **Custom** (a square that sends a specific cue).

Masters and Speed Masters

A Fader or a Button does not necessarily play a Cue List: it can also carry a group's **Master** (its intensity) or a **Speed Master** (a speed shared by effects). See also Faders and Buttons.

Placing a Master or a Speed Master

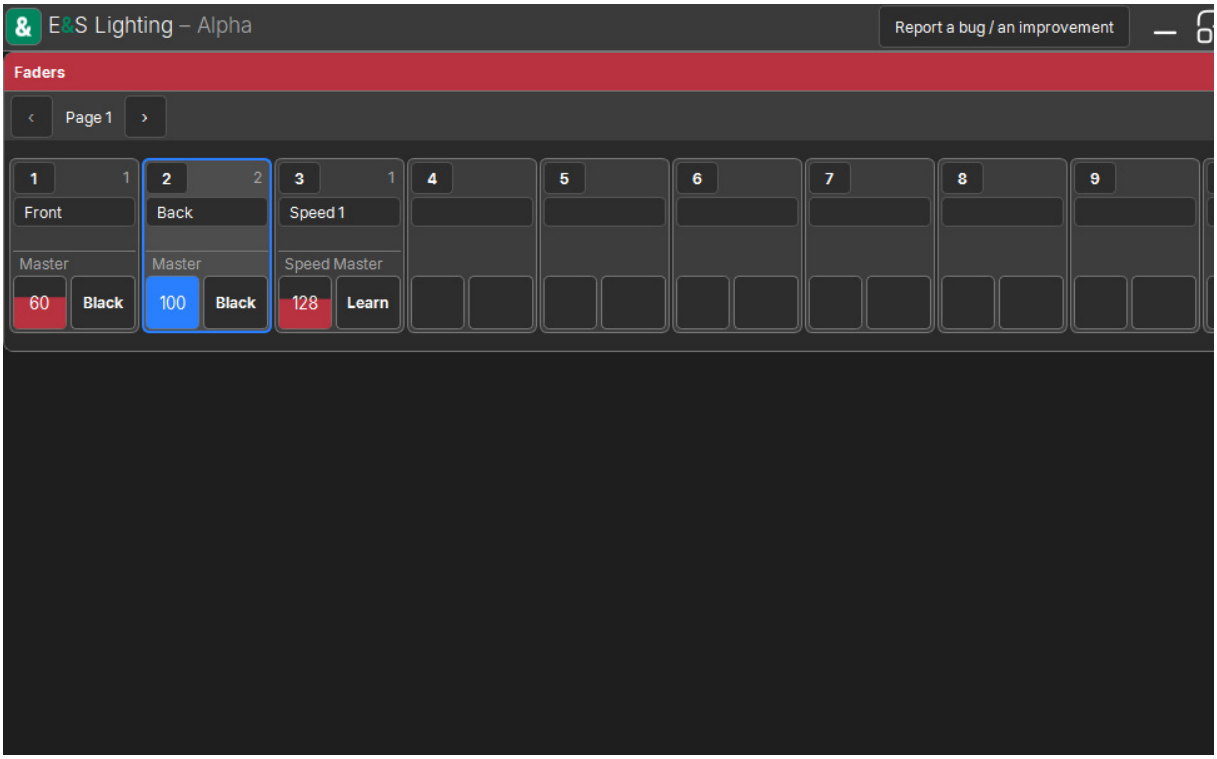


1. In the **Faders** or **Buttons** window, tap the **number** of an **empty** slot. The settings open.
2. Choose **Group master** (then the group) or **Speed Master** (then which one; "**+ New**" creates one).

If several empty slots are selected, they receive groups or Speed Masters **one after the other**: Master of group 1 on the first, of group 2 on the next...

A Cue List, on the other hand, is placed with **Assign** (see Faders and Buttons).

Group Masters



The square of a Master Fader sets the **intensity** of its group's fixtures by

multiplying it: at 50%, a fixture sent at 80% comes out at 40%. This applies to everything that plays (Cue Lists, programmer).

Its button, to choose in the settings:

Function	Effect
Black	at 0 while pressed
Flash	at full while pressed
Toggle	cut / restored at each press

The Grand Master

There is no separate Grand Master: make a **group of all the fixtures** and put its Master on a Fader. It acts as a Grand Master.

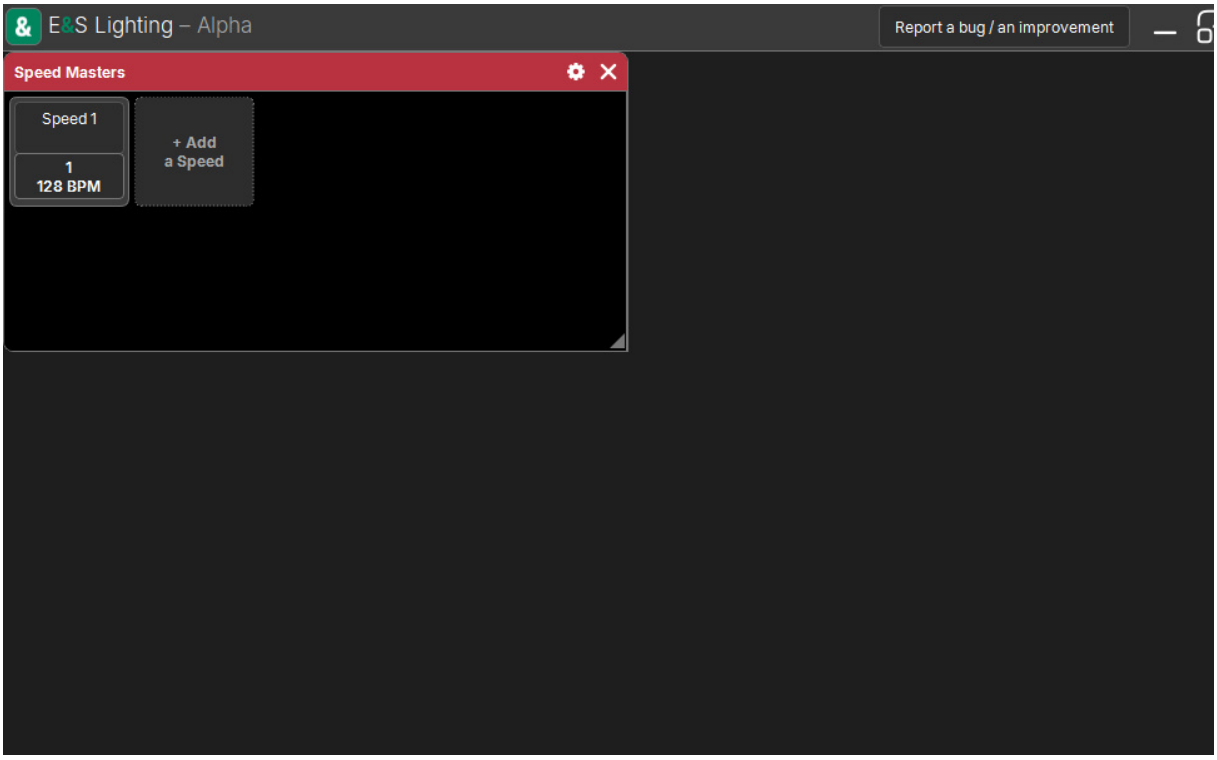
A Master only acts on intensity: a fixture **without a dimmer** (nor a virtual dimmer in its profile) is not dimmed, as on grandMA3.

A Master with no slot carrying it, or when the show is reopened, is at 100%.

Speed Masters

A Speed Master is a **shared speed** that effects follow instead of their own BPM: they stay in sync with each other. It only drives effects, not cue times.

The Speed window



"Add a window" → **Playback** → **Speed**. Speed Masters are shown as tiles; the square shows its number and its **BPM**.

- A tap on the square **taps the tempo** (Learn).
- "**+ Add a Speed**" creates one; **Delete**, **Copy**, **Move** follow the common procedure (see The lists).

Making an effect follow a Speed Master

In the effects editor, the **Speed** list of a line (or of the family band): **Own**, or a Speed Master. See Effects.

Setting the tempo

Command	Effect
the **square of a Speed Fader**	the BPM, within the **fader range** (0 to 225 BPM by default, adjustable)
Learn (button or tile)	the tempo is tapped: it takes the rhythm of your taps **and immediately realigns** the effects to your finger
×2, **÷2**	doubles or halves the BPM
60 BPM	resets to 60
Scale (settings)	multiplies or divides its speed, from ÷4 to ×4

- Learn: the first tap of a series (after more than 2 seconds without a tap) restarts from the first step, each following tap moves forward one beat.
- ×2, ÷2 and 60 BPM can go outside the fader range; moving the fader brings it back.
- Changing speed does not make effects jump: only Learn realigns them.
- Effects with **Synced** start mode on a Speed Master: they follow its beats, together.

The name and the Scale go through the history; the BPM is a setting of the moment, kept in the show.

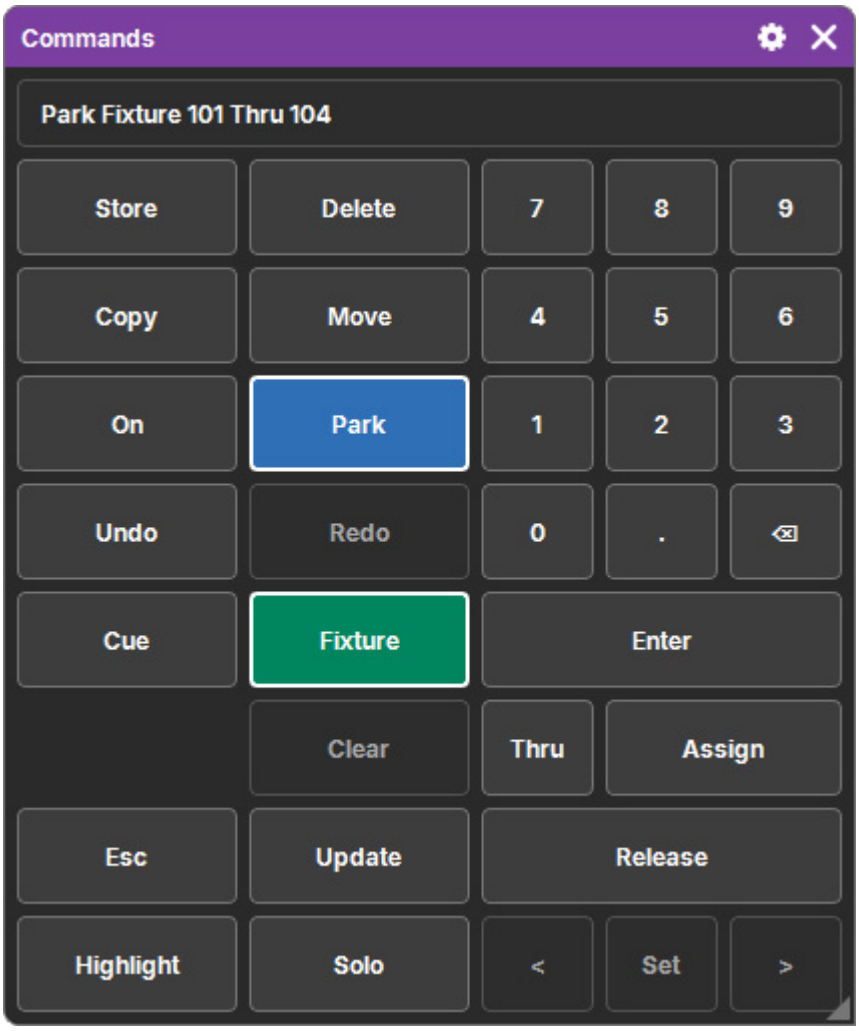
Several slots at once

Select several slots (rectangle with one finger) then tap the number of one of them: the settings apply to **the whole selection**, and only show what they have in common. The color, the group, the Speed Master, its Scale and the fader range can be set in one go this way.

Park

Park freezes fixtures, as on grandMA: they stay exactly at what was being output when you parked them, **all their attributes**, whatever happens next. Useful to keep a profile on during a focus, or to lock a faulty fixture.

Parking



On the command keypad, **two quick taps on Off**: the command line shows **Park**. Then, as you prefer:

- "**Fixture** 1 **Thru** 5" then **Enter**: these fixtures;
- **Enter** directly: the fixtures **selected** in the programmer;
- **two quick taps on Fixture** ("Park Selection") then **Enter**: the selection too.

A sub-fixture parks **its whole fixture**, head and sub-fixtures included.

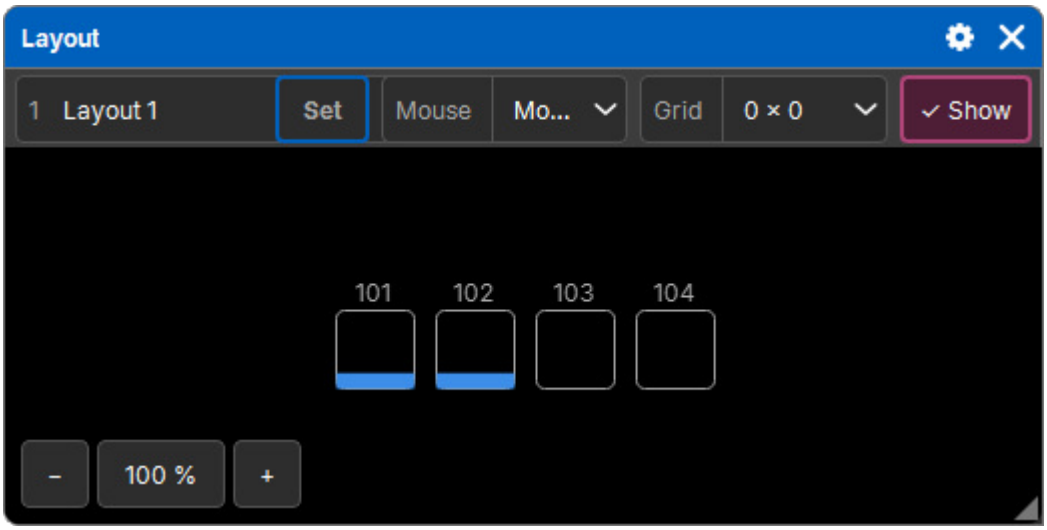
Releasing: Unpark

Two quick taps on On: the command line shows **Unpark**, then the same ways of designating the fixtures. They immediately take back what the Cue Lists and the programmer give them.

What Park does

- A parked fixture is output frozen: **nothing goes above it**, neither a Cue List, nor the programmer, nor a Cue List with Super priority.
- Its values are the ones that were being output at the time of the Park, attribute by attribute.

Where to see it



- in the **Layouts**: a **blue band** at the bottom of the fixture's square (see Layout);
- in the **Fixtures** window: a blue band under its number, and its values in blue (see Fixtures).

History and show

Park and Unpark are history entries ("Park Fixture 1 Thru 5"): **Undo** cancels them. Parked fixtures are **saved in the show**: they stay parked when it is reopened.

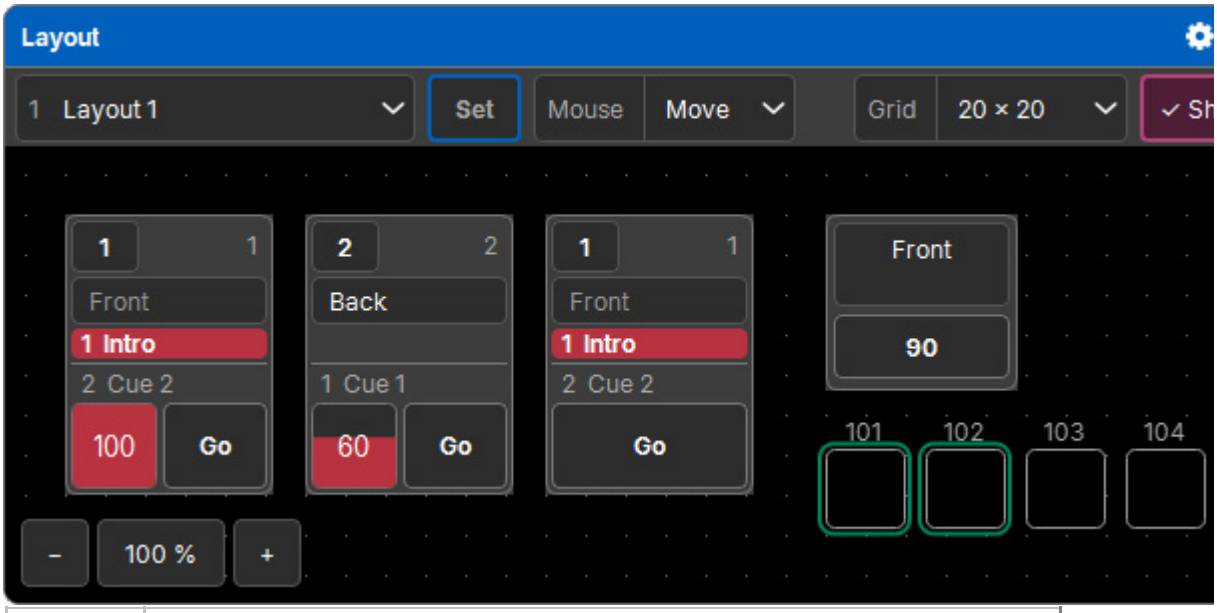
If none of the targeted fixtures is patched (or, for Unpark, none is parked), nothing happens.

Layout

A **Layout** is a custom-made panel (the Layout of grandMA): you place wherever you want **fixtures**, which show their color live, and show **objects** (groups, presets, views, Worlds, Faders, Buttons) which keep their look and their gestures. It is both a live lighting plot and a custom desk.

The **Layout** window is in the **Display** category. Layouts form a numbered pool, kept in the show.

The header



Element	Role
layout list	the layout shown (number and name); then **+ New layout**, **Rename…**, **Delete…** (on the one shown)
Set	on: you move and set up objects; off: you use the layout
Mouse	**Move** (a drag on the background moves the view) or **Select** (a drag draws a selection rectangle)
Grid	the grid step, from 0 × 0 (free placement) to 100 × 100 in steps of 10
Show	shows or hides the grid dots

+ New layout asks for **2D** or **3D** (see below). The layout shown, Set, the mouse mode and the grid display are window settings (kept in Views); the grid and the placed objects are in the show.

Zoom and panning

- In the bottom-left corner, **– 100 % +** zooms; a tap on the percentage goes back to 100 %, centered on the objects.
- **Ctrl + wheel** zooms around the mouse.
- On a touchscreen, **two fingers** pan and zoom the view, in all modes, without ever touching what is underneath.

- With the mouse in **Move**, a drag on the background pans the view.

The zoom and the area viewed are remembered by each layout: you find them again when you come back to it, in any window.

Placing fixtures

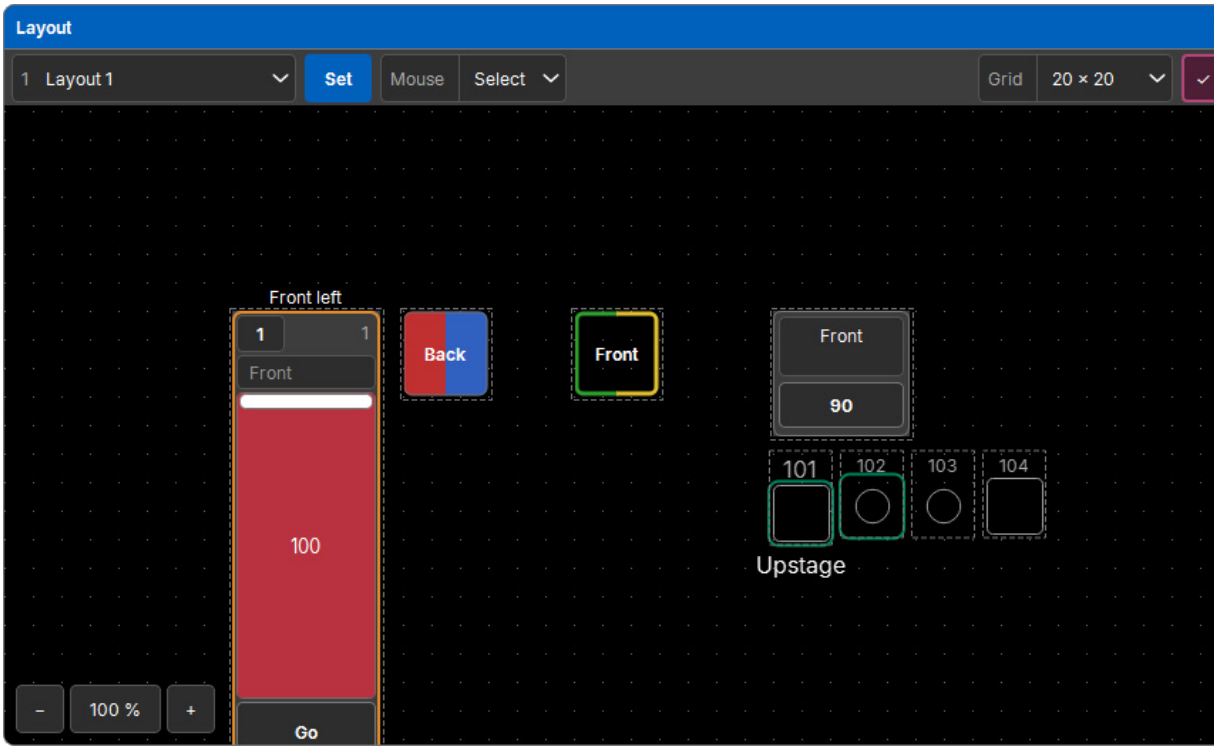
1. Select fixtures in the programmer (by hand, with a group...).
2. **Store**, then a tap on the layout background.

The fixtures are placed in selection order, starting from the tap, side by side in rows. A fixture with sub-fixtures (an LED bar, for example) gets its own row: its **head** first ("101", a tap selects the whole fixture), then its sub-fixtures in line (101.1, 101.2...).

Each fixture is a square with rounded corners:

- **in the color it outputs**, darkened according to its intensity, live (several colors, for example a two-color wheel: in stripes);
- its number above it;
- selected in the programmer: a frame in the programming color.

Placing objects



Assign, then an object (a group, a preset, a view, a World, a Fader, a Button), then a tap on the layout background: it is placed there, as it is in its window, with the same gestures. Several selected objects are placed at once, in rows.

A placed Fader or Button keeps **its page**: it does not follow the page shown in the Faders and Buttons windows.

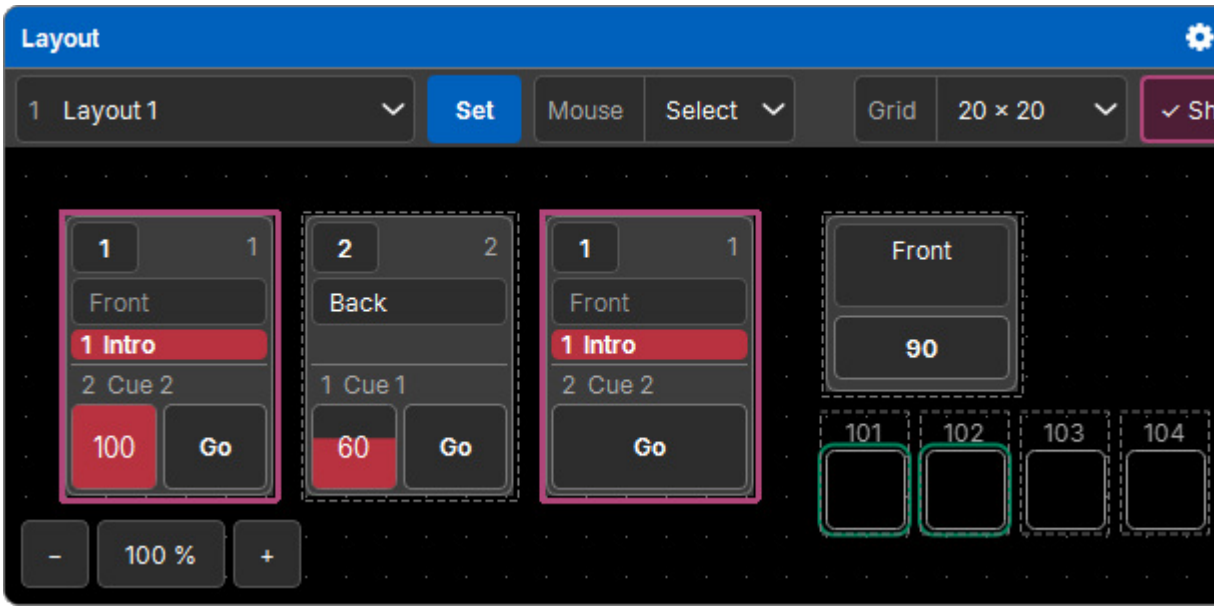
Using the layout (Set off)

- Placed objects react **as in their window**: a group selects, a preset applies, a

Fader slides...

- A tap on a **fixture** selects it in the programmer, or deselects it if it was selected. On the head of a fixture with sub-fixtures: the whole fixture.
- Mouse **Select**: a rectangle drawn on the background adds the fixtures it touches to the selection, **in the order of the drawing** (row by row, in the direction of the drag). Their place in the selection grid follows their arrangement in the layout: useful for effects and the Pattern.
- **Off** then a fixture: it leaves the programmer and the selection.
- **Delete** then a placed object: it is **removed from the layout** (the object itself, group or preset, stays in the show).

Arranging and setting up (Set on)



In **Set** mode, objects no longer react: you arrange them. They have an outline, solid and colored when they are selected.

- **Dragging an object** moves it, by grid steps, with the whole selection if it is part of it. Near the edge, the view scrolls toward it.
- Mouse **Select**: a rectangle on the background selects objects; a tap on an object selects or deselects it. **Esc** clears the selection.
- **Long press** on an object: its settings (below).
- **Copy** then an object (or the selection), then a tap on the background of a layout (the same or another one): the copies are placed there.
- **Delete** removes from the layout the object pressed, or the whole selection if it is part of it.

Position, size and arrangement

When objects are selected in Set mode, the **Programmer** window shows their settings instead of the families, in pages of tiles:

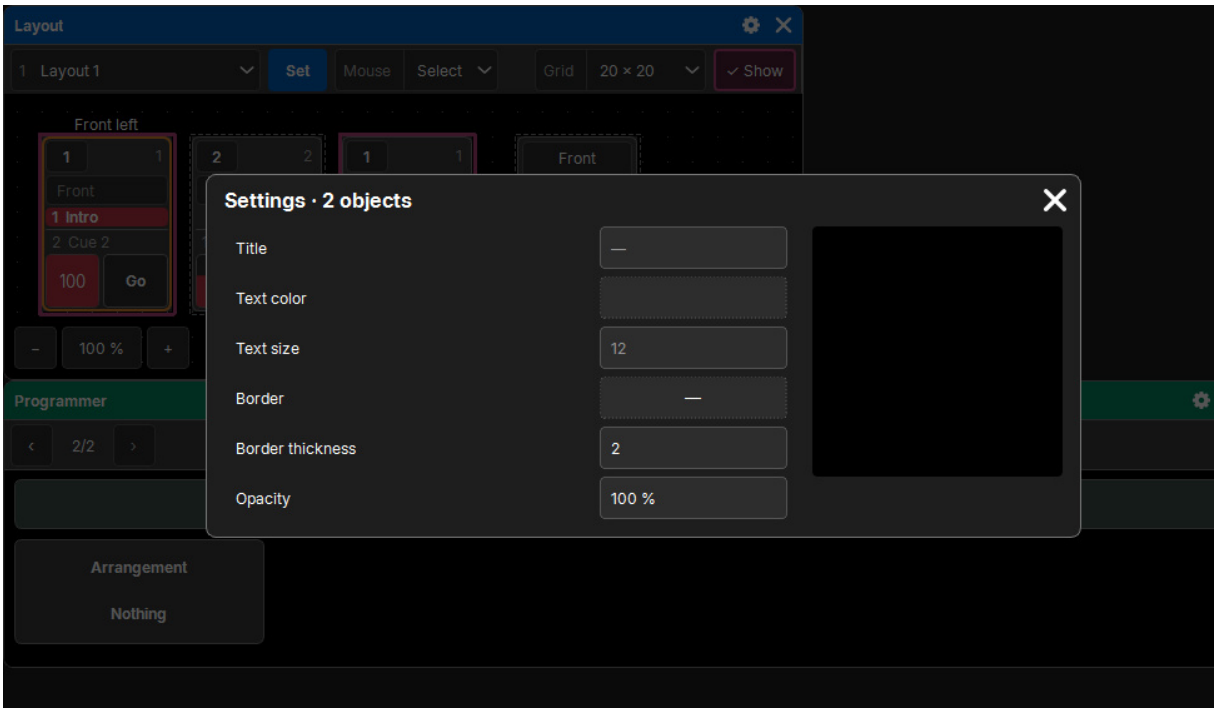
- **X**, **Y** (the top-left corner), **Width**, **Height**. A tile shows the value common to the objects (empty if they differ); a typed value goes to all of them; "100 thru 500" is spread in selection order;

- the **Arrangement**:

Arrangement	Effect
Nothing	each object at its original place
Line	evenly spaced on a line of a given length and angle (0°: to the right, 90°: downward); its **Origin** says whether it starts from the objects' starting point, is centered on it or ends at it
Grid	row by row, a number of columns, a margin between cells
Circle	evenly on a circle of a given diameter, the first one at the top, turned by an angle
Rotation	the original places turned by an angle (clockwise), all together or fixture by fixture (a fixture and its sub-fixtures around their own point)

The arrangement always starts again from the original places: **Nothing** puts them back. Each change is a history step.

The settings popup



A long press on an object opens its settings (those of the whole selection if it is part of it). On the left, the categories; at the top right, a preview.

- **General** (any object): a **title** above the object (below it for a fixture), the text color and size (title and number of a fixture), the **border** color and thickness, the **opacity**.
- **Appearance** of fixtures: an **image** from the Images pool instead of the square (the frame takes its proportions), and the **source** that shows the color live: its shape (**Rectangle** or **Circle**) and its place as a % of the frame. You can also drag it and pull its corners in the preview.
- **Appearance** of Faders and Buttons: **Normal** (the slot of the Faders window),

Large (number, name, a large fader with a handle and the button below), or **Custom** (a square, see below).

A field showing "—" has different values depending on the objects; a value given goes to all of them.

Custom squares

A Fader or a Button with the **Custom** appearance becomes a square, set up in the **Action** category:

- **Activation Simple**: a tap does what its button does (its function: Go, Flash...). Active when its Cue List is playing.
- **Activation Cue**: it sends **a specific cue** of its Cue List. Active when this cue is the current cue.
    - **Mode**: **Go** (with its times), **Flash** (while pressed, without times), **Temp** (while pressed, with times) or **Toggle** (one tap sends it, the next one goes back to what was there before);
    - **Return to**: on release (or on the second tap), the **Previous cue**, or **Off** (the Cue List goes off).

Its look, **active** and **inactive**: a border (thickness, colors), a background (colors, several in stripes, or none), an image from the pool and its size; common to both: the corner radius, the text (the cue's name, otherwise its own; it can be hidden) and the brightness.

Automatic (in Cue): the color or the image comes from the cue itself. The color the cue gives its fixtures (Color attribute), or the image of the value it gives an image-wheel attribute. A color desk builds itself this way.

3D layouts

A layout created in **3D** only contains fixtures, placed in space.

- It is viewed from one side at a time, like a 2D layout: **Front**, **Back**, **Left**, **Right**, **Top**, **Bottom**. The third axis is depth.
- A **layer** is a plane at a given depth: you place and move fixtures on it as in 2D. The list shows **All** layers, each depth that has fixtures, and **+ New layer** (further than the last one). **<** and **>** go to the previous or next layer.
- These controls are in the bottom-right corner of the area ("Front ▾ < Z 100 ▾ >").
- In Set mode, the position settings also have a **Z**; arrangement happens in the plane of the side viewed.
- The gear sets the opacity of fixtures on other layers (0: hidden).
- The **3D** button shows the scene in volume: a box per fixture (a sphere for a round source), in its live color, on a grid floor. Dragging orbits around, two fingers (or right / middle click) pan, pinching (or the wheel) zooms in, a double tap reframes. This view is for looking: placement happens in the sides.

Worlds

The window's gear sets the **opacity of fixtures outside the active World**: they stay visible, grayed out (see Worlds).

License

Without an active license, a gray veil covers the layouts with the **Activate the license** button (see License).

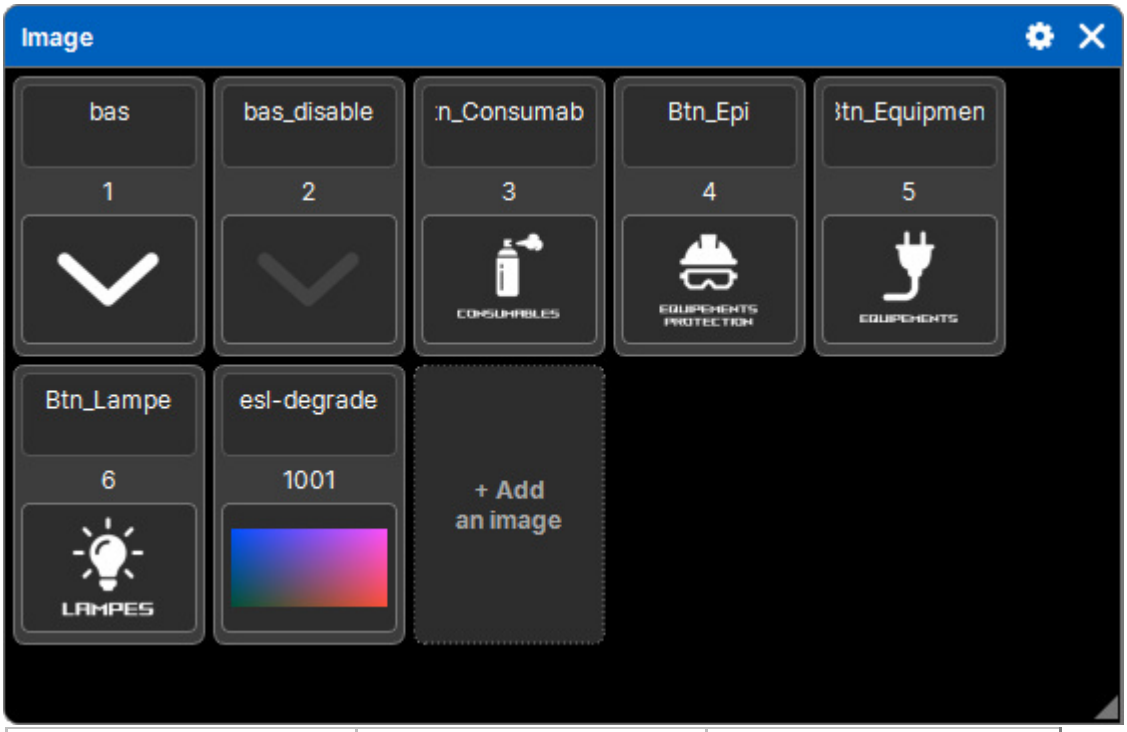
A large layout (thousands of fixtures) loads in batches: the interface stays usable during loading, a wheel spins until it finishes.

Images

The **Image** pool keeps the images used by Layouts: the image of a placed fixture (instead of its square), or the background of a custom Fader or Button square.

The **Image** window is in the **Display** category. Images are shown as tiles there: the name, the number, the thumbnail.

Two kinds of images



Kind	Numbers	What you can do with them
Base images, supplied with the software	from 1 to N, in the order of their names	view them (a tap shows them full size) and **copy** them into the show; never edit or delete them
Show images	after the base images	replace, rename, delete, copy, move them

Base images are updated with the software; show images are saved in the show file, which stays complete when you move it to another computer.

Adding images

+ Add an image opens the file browser; you can choose several files at once.

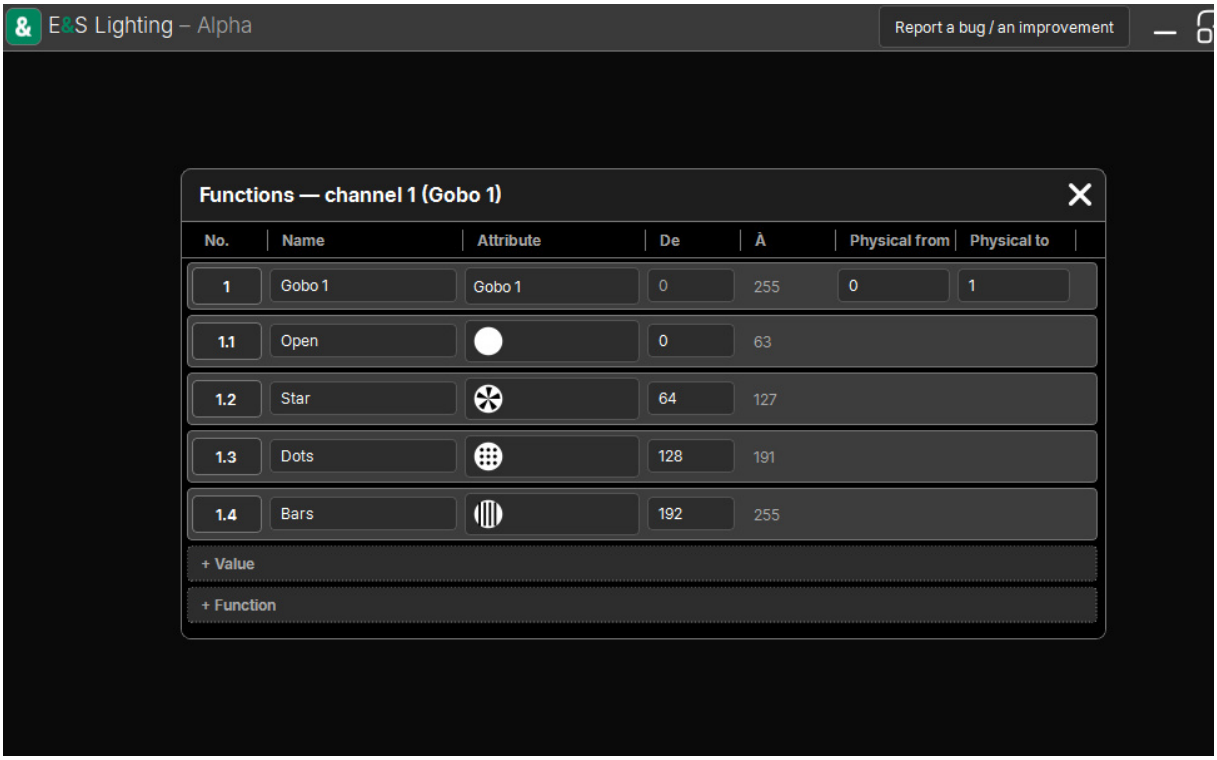
- Formats read: PNG, JPG, BMP, SVG and GIF (still image).
- A photo is rotated according to its orientation (the camera's EXIF data), then reduced to **1024 pixels** on its longest side: a show does not get weighed down by a 20 MB photo.

- It is kept as PNG if it has transparency, otherwise as JPG.
- An unreadable file is reported in the command line ("Unreadable image: ..."); the others are added.

Replacing, renaming, organizing

- A tap on a **show** image (with no pad key) opens the file browser to **replace** it: wherever it is used, the new one takes its place.
- Its name is typed directly in its tile.
- **Delete**, **Copy**, **Move** follow the common sequence (see Delete, copy, move). Nothing can arrive on the number of a base image.

Profile images



The images of **gobo wheels** (or animation wheels) are not in this pool: they are stored with the values of the fixture's profile, and are set in the Profiles window. They show in the ready-made values of the Programmer, and in a Layout square with the Automatic appearance.

Clock

The **Clock** window (**Display** category), like Clock on grandMA, shows a time in large digits, at the size of the window: the computer's time or a timecode. It is also used to control the **Internal** timecode.

What it shows

A list, at the top, chooses the source:

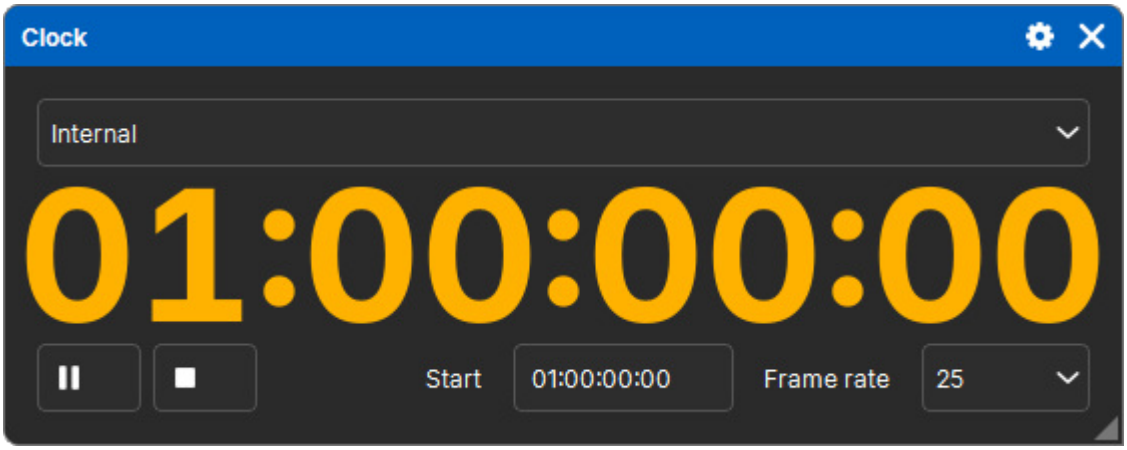
Source	Display
Current time	the computer's time, `HH:MM:SS`
Internal	the software's Internal timecode, `HH:MM:SS:FF`
a **Time input**	the timecode received on that input (see Timecode)

As long as there is no timecode (silent input, Internal stopped), the clock shows `- - : - - : - - : - -`. An input deleted since shows "... not found".

The text takes all the available space, in width or height; the digits all have the same width, so the size does not jump from one frame to the next.

The source and the **text color** (in the window's gear) are remembered with the window, and in Views. Several clocks can show different sources.

The Internal timecode



The software has its own timecode, the **Internal** one: it counts time from Play, starting at a start time. It serves as a source for Timecodes without plugging anything in: to rehearse a show timed to the clock, or for a show that runs on its own.

When the clock shows **Internal**, its controls appear:

Control	Effect
Play	time runs; the button becomes **Pause**
Pause	time stops where it is; Play starts it again from there
Stop	no more timecode; the next Play starts again from the start time
Start	the start time, `HH:MM:SS:FF`

Frame rate	24, 25, 29.97 or 30 frames per second

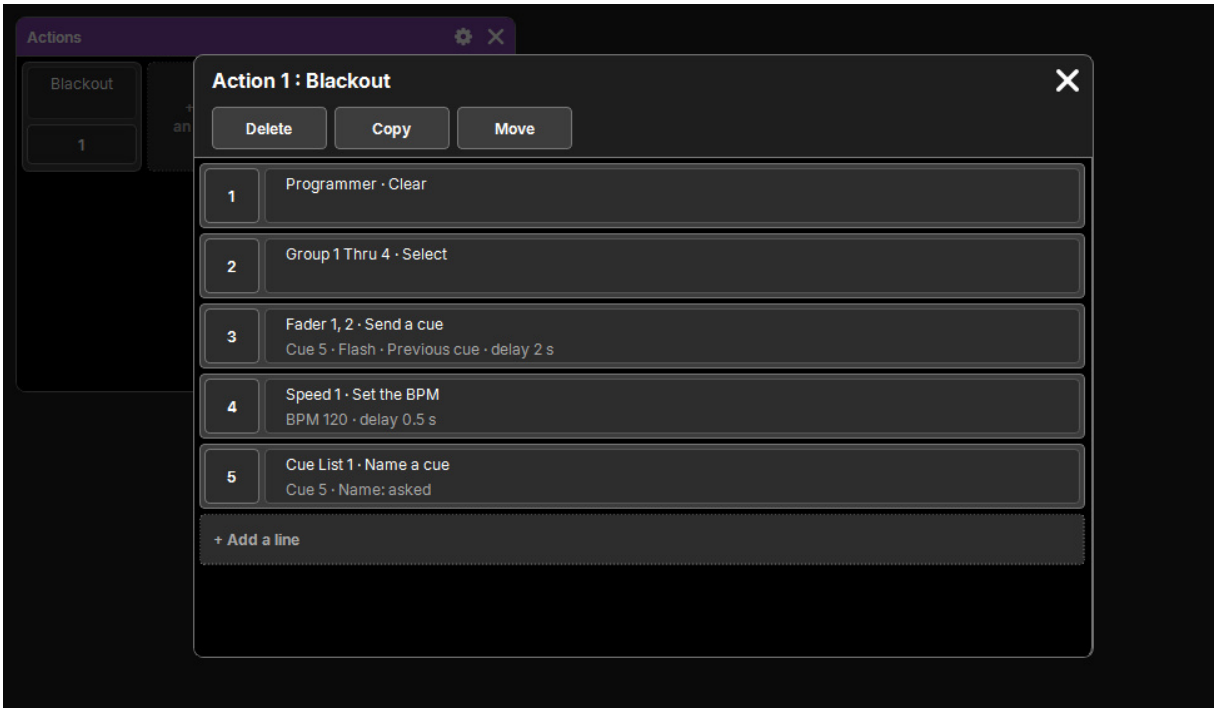
- There is only **one** Internal timecode: all clocks control it together.
- The start time and the frame rate are kept in the show. Playback, however, is the current state: when the show opens, the Internal timecode is stopped.
- Starting again with Play only triggers the Timecode lines at the exact start instant, not those before it (see Timecode).

Actions

An **Action** is a series of commands played with a single gesture, like a grandMA macro, but with nothing to write: each line is built in a menu. An Action can start cues, set Faders, select groups, recall a View, change page...

The **Actions** window is in the **Control** category: the pool of Actions, as tiles.

Running and setting up



- **A tap** on an Action **runs** it. A new Action is empty.
- **A long press** opens its **settings**.
- **+ Add an Action** creates a new one; its name is typed in its tile.
- **Delete**, **Copy**, **Move**: the common pool sequence (see Delete, copy, move).

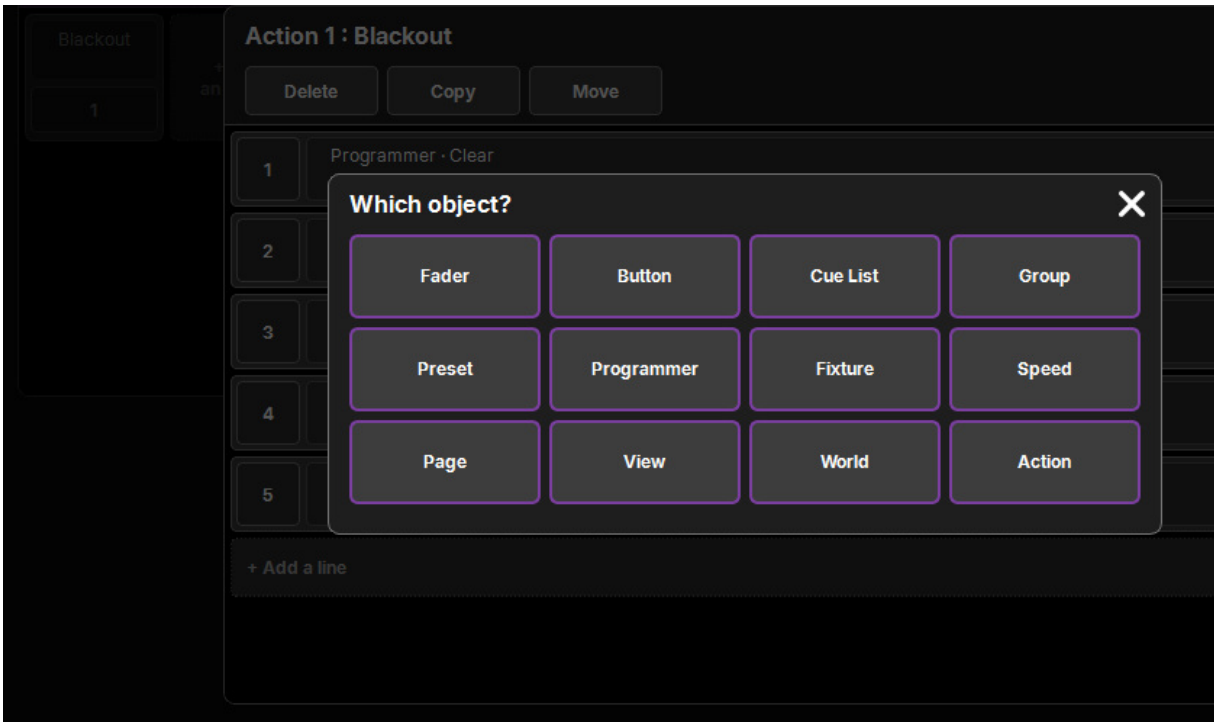
An Action can also be run from a **DMX Input line** (a channel received, see DMX Input), a **Timecode line** (see Timecode), or another Action.

The lines

The settings popup shows its **lines as numbered cards**, played in order. Each card summarizes the line on two lines: the object and the action, then its parameters.

- **+ Add a line** creates a new one.
- A tap on a card opens its settings.
- The popup covers the screen area, so sometimes the pad: its **Delete**, **Copy** and **Move** keys are repeated at the top, and do exactly the same as those of the pad.

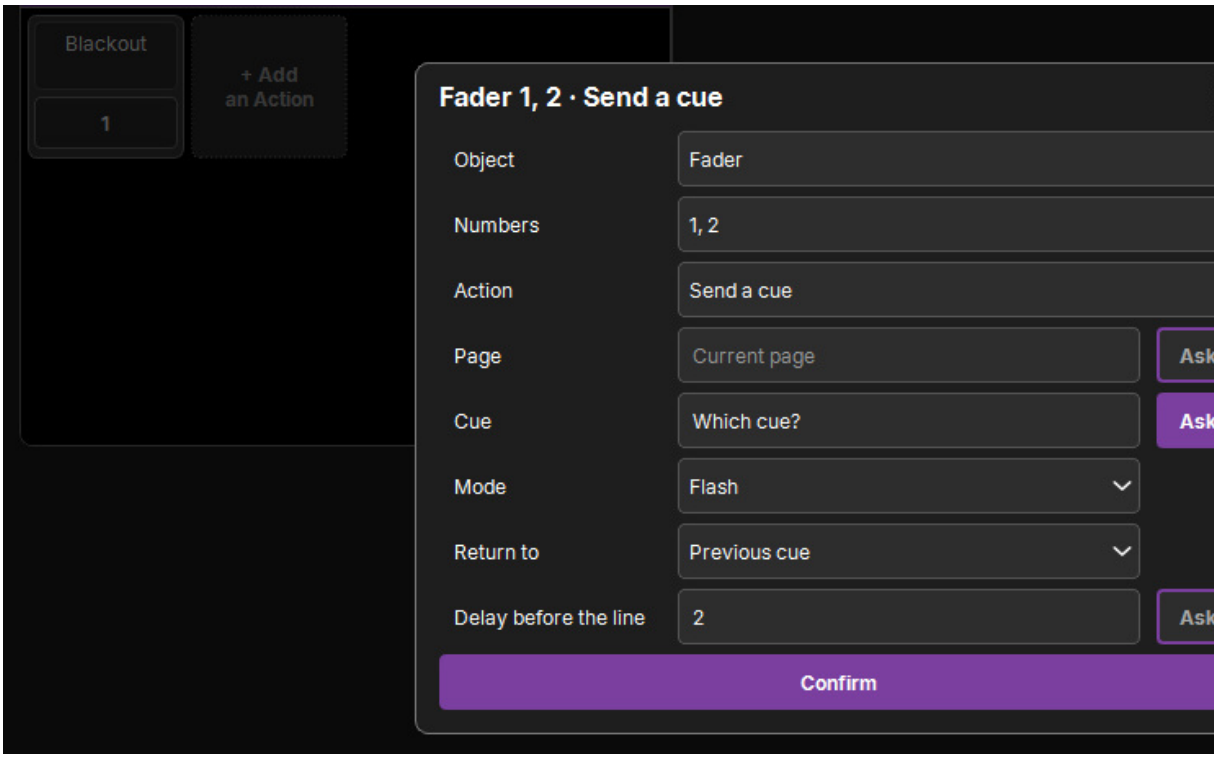
Building a line



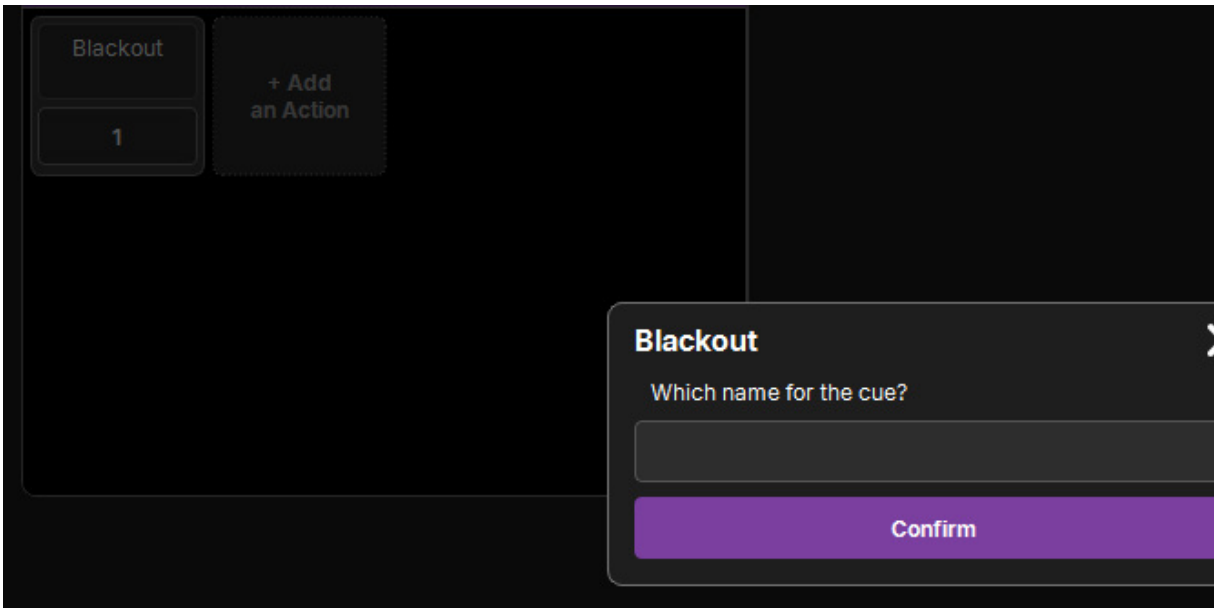
In four steps, with **‹ Back** to return to the previous step:

1. **Which object?** — the type of object: Fader, Button, Cue List, Group, Preset, Programmer, Fixture, Speed, Page, View, World, Action.
2. **Which ones?** — one or more objects, as tiles: a tap picks or removes an object, in the order of the taps (the order matters for a fixture selection). Or the **Numbers** field with the keyboard: `1, 3, 5 thru 8`; for fixtures, as on the pad: `1, 4.1 thru 403.12`. An object that does not exist yet is accepted. For a Fader or a Button, the **page** is chosen here (empty: the current page at run time); for a Preset, its **family**. No step for what has no number (Programmer, Page).
3. **Which action?** — what the line does (the catalog, below).
4. **The parameters** of the action (a cue, a value, a name…) and the **Delay before the line**, in seconds (empty: none).

Confirm writes the line, in a single history step; the cross cancels everything. Editing an existing line opens the parameters step directly.



Ask at run time



Each number or text of a line, **objects included**, can be **asked at run time**, with its question ("Which cue?"). At run time, the Action stops at this line and a popup asks the question on the main screen, even if the Action was started by a DMX Input. The question's cross **cancels the rest** of the Action.

A "Go to a cue..." Action on Cue List 1 that asks for the number thus makes a "go to any cue" button.

The catalog

Object	Actions
Fader, **Button**	**Press**, **Release** (the slot's button, with its function), **Level** (Fader only), **Go**, **Go−**, **Top**, **Pause**, **Off**, **Send a cue** (a cue, its **Mode** Go, Flash, Temp or Toggle, and **Return to**: Previous cue or Off; a Release line on the same slot brings it back), **Store** (a new cue, like **Store** then the button)
Cue List	**Go**, **Go−**, **Top**, **Pause**, **Off**, **Go to a cue**, **Store a cue** (empty: a new one, at the end), **Name a cue**
Group	**Select**, **Add to selection**, **Off**, **Store**, **Name**
Preset	**Recall**, **Off** (what comes from it leaves the programmer), **Store**, **Name**
Programmer	**Clear**, **Clear all**, **Set a value** (an attribute, a value), **Highlight**, **Solo** (on or off), **Next fixture**, **Previous fixture**
Fixture	**Select**, **Add to selection**
Speed	**Set the BPM**, **×2**, **÷2**, **Learn**
Page	**Change page**
View	**Recall**
World	**Activate**
Action	**Run**

Storing from an Action

Store (a cue, a group, a preset) does the same storing as on the pad: a free number creates the object. If it already exists, the **If it already exists** parameter chooses:

- **Ask**: the Replace / Merge / Remove popup, as on the pad;
- **Replace**, **Merge** or **Remove**, without asking anything: for an Action run with nobody in front of the screen (DMX Input, Timecode).

Running

- Lines **with no delay** start together; a line's delay counts from the end of the previous one.
- Several Actions can play at the same time. **Running** one that is playing again **restarts it from the beginning**.
- An Action that would run itself, directly or through another, is **refused** (a message says so): no endless loop.
- The lines are copied at run time: editing an Action that is playing does not change what is running.
- Running an Action does not go into the history; only what a line changes in the show does (Store, Name).

An Action only keeps numbers. If you move a Cue List, a cue, a group or an Action referred to by a line, the line follows the new number. Only the fixtures typed in the Numbers field stay as written: they are read again at run time, with the patch of that moment.

DMX Input

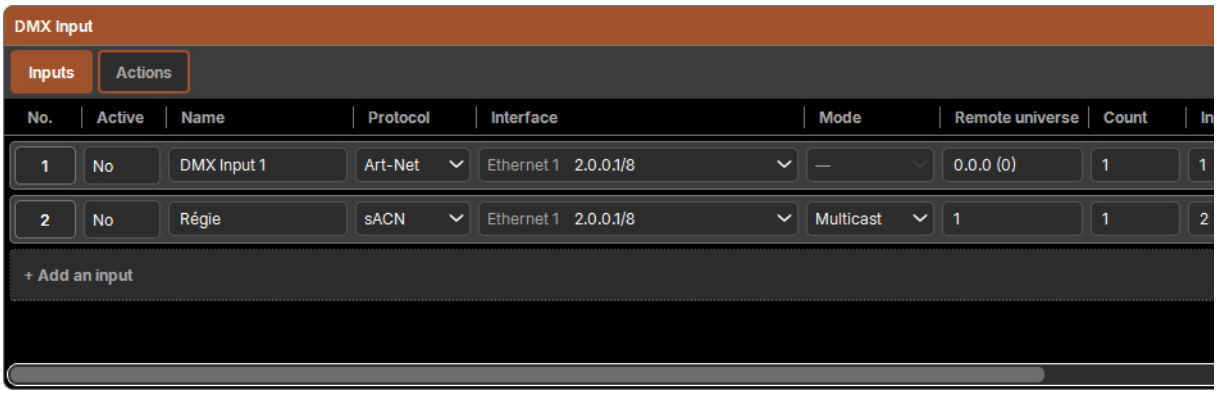
DMX Input lets DMX received from the network trigger something in E&S Lighting: raise a Fader from another console, fire a cue from a DMX button, run an Action... It is the equivalent of grandMA3's DMX Remotes. **Nothing that is received goes back out.**

The **DMX Input** window ("Add a window" → **Setup**) has two categories at the top, remembered with the window:

- **Inputs**: what is listened to, and what actually arrives;
- **Actions**: what it triggers, and what each channel receives.

Settings are saved in the show and go through the history; triggers do not (like a tap by hand).

Inputs



One row = universes listened to on a network card, received under a separate numbering: the **input universes**, unrelated to the patch.

Column	Purpose
No.	the pad object (Delete, Copy, Move)
Active	**Yes** / **No**
Name	"DMX Input 1" by default
Protocol	**Art-Net** or **sACN**
Interface	the network card that listens (as in DMX Output); a missing card stays shown, "not found"
Mode	sACN: **Multicast** (subscribe to each universe's group) or **Unicast** (receive what is sent directly to the computer); Art-Net: nothing to choose
Remote universe	the first universe received from the network (Art-Net: Net.Sub-Net.Universe or absolute, from 0; sACN: from 1)
Count	how many universes in sequence
Input universe	the number under which the first one is received (from 1 to 63,999)
Status	what arrives: "Received from 192.168.1.20" ("+1" if there are

	other sources), **No signal**, or "—" for an inactive input

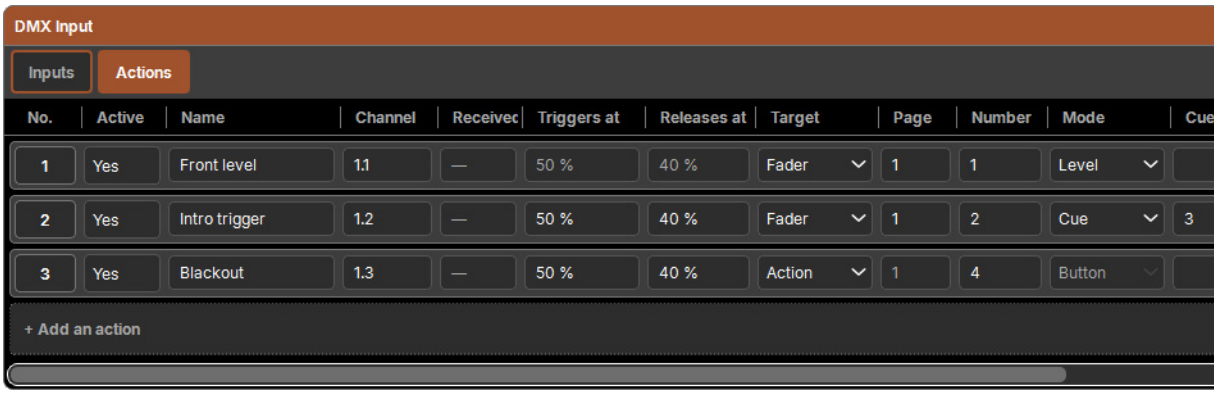
+ Add an input adds one after the last one (same settings, universes right after); it starts **inactive**, like an output.

Several sources

- **sACN**: the highest priority wins; on a tie, the highest value of each channel.
- **Art-Net**: the highest value of each channel.
- A source that goes silent is forgotten after **10 s** in Art-Net, **2.5 s** in sACN, or as soon as its stream ends.
- sACN Preview data is ignored.

The software's replies to ArtPolls announce the universes it listens to: a console in unicast (or in Auto) knows where to send them.

Actions



One row = a channel listened to and what it triggers.

Column	Purpose
No.	the pad object
Active	**Yes** / **No**
Name	free
Channel	`input universe.channel`: `1.12` = input universe 1, channel 12 (8 bit)
Received	what the channel receives, in %; "—" with no signal
Triggers at	the press threshold, 50 % by default
Releases at	the release threshold, 40 % by default
Target	**Fader**, **Button** or **Action**
Page	the page of the Fader or Button
Number	the number of the Fader, Button or Action
Mode	**Level**, **Button** or **Cue**
Cue	the cue to send (Cue mode)
Cue mode	**Go**, **Flash**, **Temp** or **Toggle** (Cue mode)

Return to	**Previous cue** or **Off**: where the Cue List goes back to on release (except in Go)

Only the useful cells stay active: Level has no thresholds, an Action has neither page nor mode, the cue mode only exists in Cue mode.

+ Add an action adds one after the last one, on the next channel.

The page is fixed

The target is a Fader or a Button **on a given page**: changing page on the Faders and Buttons does not change what the DMX triggers.

The three modes

- **Level** (Fader): the channel **is** the Fader's level. It only applies when the received value **changes**: you can still move the Fader by hand between two movements of the channel. A Button has no level.
- **Button**: the channel goes above **Triggers at**: the button is **pressed**, with its function (Go, Flash, Temp...); it goes back below **Releases at**: **released**.
- **Cue**: the same, but sends a specific cue of the Fader's or Button's Cue List, in **Go**, **Flash**, **Temp** or **Toggle**, with its **Return to** (like a Custom square in a Layout).

The gap between the two thresholds prevents a channel hovering around one value from triggering over and over.

An **Action** target runs the pool's Action when the threshold is crossed.

No signal

No more signal, an action turned off, changed or removed: what it was holding **pressed is released**, so that a Flash does not stay on. A level, on the other hand, stays where it is.

Selection and sequential entry

As in DMX Output: a rectangle, or a column title for the whole column, then a typed value and **Enter**:

- what follows on goes **in sequence**: remote and input universe, channel, target number (`1.1`, `1.2`, `1.3`...);
- the rest takes **the same value**.

A choice (Active, Protocol, Target, Mode...) made in a selected cell applies to the whole selection.

Timecode

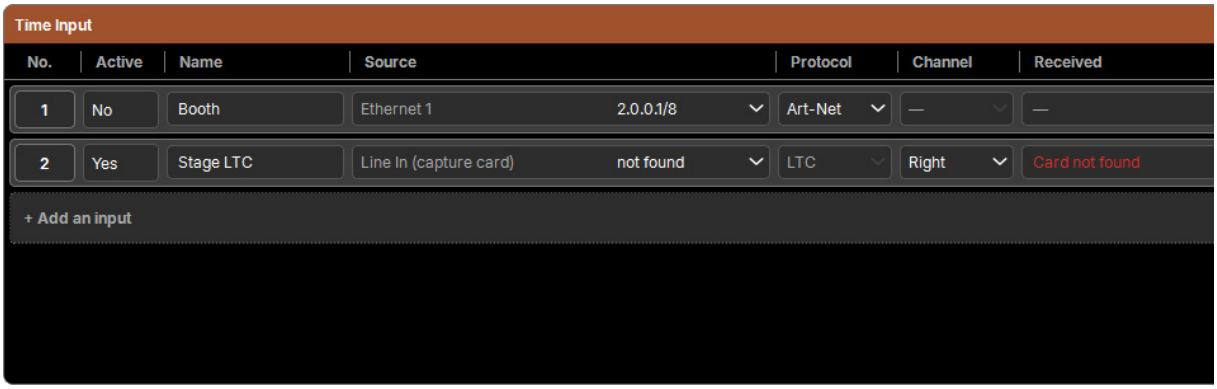
Timecode starts **cues** and **Actions** at a precise time, coming from outside (a sound desk, a video control room, an LTC track) or from the software itself (the **Internal** timecode).

Three windows:

- **Time Input** (Setup): what is being listened to;
- **Timecode** (Playback): what starts, and when;
- **Clock** (Display): the time in large digits, and the Internal timecode controls (see Clock).

A time is written `HH:MM:SS:FF` (hours, minutes, seconds, frames). Times are compared as they are displayed: two sources with different frame rates that show the same time are at the same instant, nothing drifts.

Time Input: what is being listened to



One line per source, with **+ Add an input**. A new input is created **inactive**: set it up, then switch it to **Active**.

Column	Content
No.	its number, referred to by Timecodes and Clocks
Active	Yes / No
Name	a free name
Source	a **network card** (Art-Net timecode) or a **sound card** (LTC)
Protocol	network: **Art-Net**; sound card: LTC, nothing to choose
Channel	sound card: **Left** or **Right**
Received	the time read right now, or what is missing

Network: Art-Net

Art-Net timecode (ArtTimeCode) arrives on the chosen network card, on the usual Art-Net port. MTC usually goes through MIDI, which is not supported yet.

Sound card: LTC

LTC (audio timecode) is read by the chosen sound input, on the left or right channel, and decoded by the software, forward as well as backward. Available on

Windows and Linux.

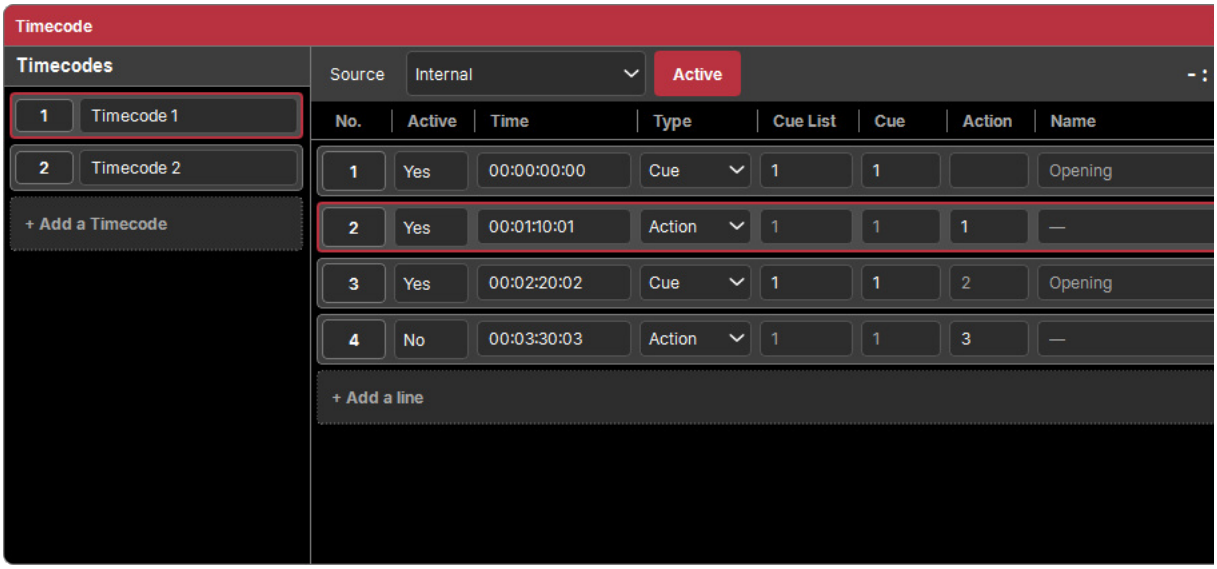
- A card is only opened if an **active** input asks for it.
- Sound cards plugged in or unplugged are followed: a card that disappears is reported as **Card not found** and kept in the settings; it is picked up again automatically when it comes back.

The Received column

Display	Meaning
`01:02:03:04`	the time received
No signal	nothing on the network for this input
No sound	the sound card receives nothing
Sound received, no LTC	it hears something, but not a timecode, or its level is too low: check the wiring and the gain
Card not found, **Card unavailable**	the card is not on this computer, or the system refuses it (used elsewhere, permissions)
—	the input is not active

In red: a problem to fix. With nothing received for **1 second**, there is no more timecode on this input.

Timecode: what starts



Like the Cue List: on the left the **Timecodes**, on the right the **lines** of the one shown.

A Timecode

- Its **source**: **Internal**, or a Time input.
- **Active**: an inactive Timecode triggers nothing.
- Its live time, above the lines.
- **+ Add a Timecode** creates a new one.

You can have several: one per song, for example, each with its lines, or several

different sources.

The lines

One line per trigger, sorted by time:

Column	Content
No.	its number
Active	an inactive line does not start
Time	the instant, `HH:MM:SS:FF`
Type	**Cue** or **Action**
Cue List, **Cue**	for a cue: its Cue List and its number. The cue starts **with its times**, like a Go
Action	for an Action: its number (see Actions)
Name	the name of what will start

- **+ Add a line** takes the time running right now (otherwise `00:00:00:00`): start the timecode, add a line at the right moment, then correct the time if needed.
- The last lines that started are highlighted until the next ones, like the cue that is playing.
- **Delete** and **Copy** follow the common sequence; you do not **move** a line: its time sorts it.
- A renumbered cue or a moved Action: the lines that refer to them follow.

Triggering

At every moment, each active Timecode reads its source:

- **time moves forward**: the active lines that have passed start, each once, in order;
- **it goes back**, or **jumps by more than 2 seconds** (a rewind, a track skip): the software follows, without sending anything that was skipped, and keeps listening;
- **it comes back** after an absence (or a Play of the Internal timecode): only the lines of that exact instant start.

Triggers do not go into the history, like a press by hand.

Timecode never replays what it skipped. To resume a show in the middle, set the source to the right time, then go to the right cue in the Cue List (a tap on its number): what follows will start normally.

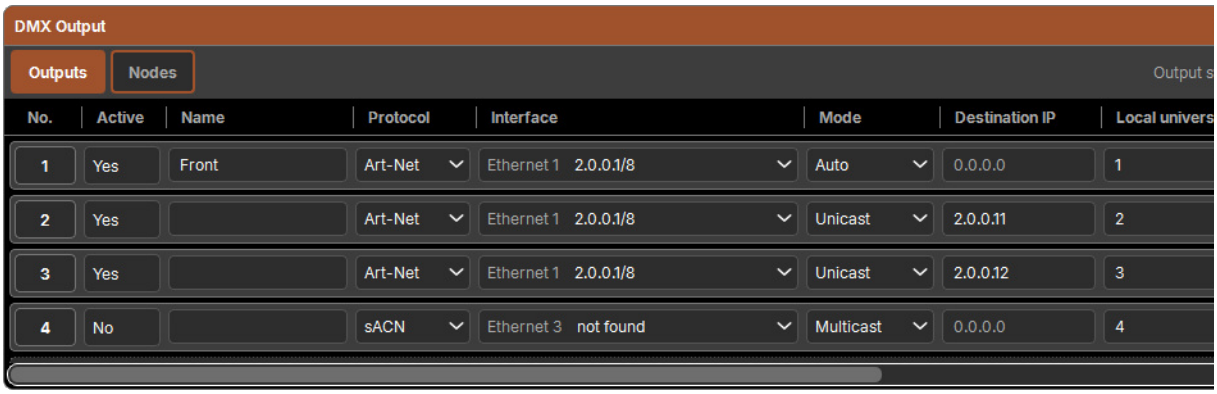
DMX Output

E&S Lighting sends DMX over the network, in **Art-Net** and **sACN** (E1.31). A node (or a fixture that receives the network directly) turns these frames into DMX on its outputs.

The **DMX Output** window ("Add a window" → **Setup**) has two categories at the top: **Outputs** (what is sent) and **Nodes** (what is seen on the network). The category shown and the column widths are remembered with the window. Outputs are saved in the show.

Attention: without an active license, no universe is sent.

Outputs



One row = a set of universes sent through one network card. Like grandMA's Art-Net and sACN tables combined: as many rows as you want (a single one in Auto is often enough; one per node in unicast on a large network).

Column	Purpose
No.	the pad object (Delete, Copy, Move)
Active	**Yes** / **No**: an inactive output sends nothing
Name	free
Protocol	**Art-Net** or **sACN**
Interface	the network card that sends (see below)
Mode	Art-Net: **Auto**, **Broadcast** or **Unicast**; sACN: **Multicast** or **Unicast**
Destination IP	the node's address, in unicast only (grayed out otherwise); `0.0.0.0`: the output sends nothing
Local universe	the first universe of the software (the one in the patch) sent by this row
Count	how many universes in sequence
Remote universe	the first universe on the network
Priority	sACN only: from 0 to 200, 100 by default
Preview	sACN only: preview data, for a visualizer (stage receivers

	ignore it)
Delay (ms)	a pause between two universes, from 0 to 100 ms, for old nodes that get overloaded

+ Add an output adds one **in sequence**: like the last one (protocol, interface, mode), with its local and remote universes right after its own. The very first one goes out through the first card, in **Auto**, universe 1 to Art-Net universe 0. A new output starts **inactive**: set it up, then switch **Active** to **Yes**.

Remote universe

- **Art-Net**: as **Net.Sub-Net.Universe** (`0.1.3`) or absolute (`19` = Net × 256 + Sub-Net × 16 + Universe), from 0 to 32,767. Both notations are accepted; the cell shows "0.1.3 (19)".
- **sACN**: from 1 to 63,999.

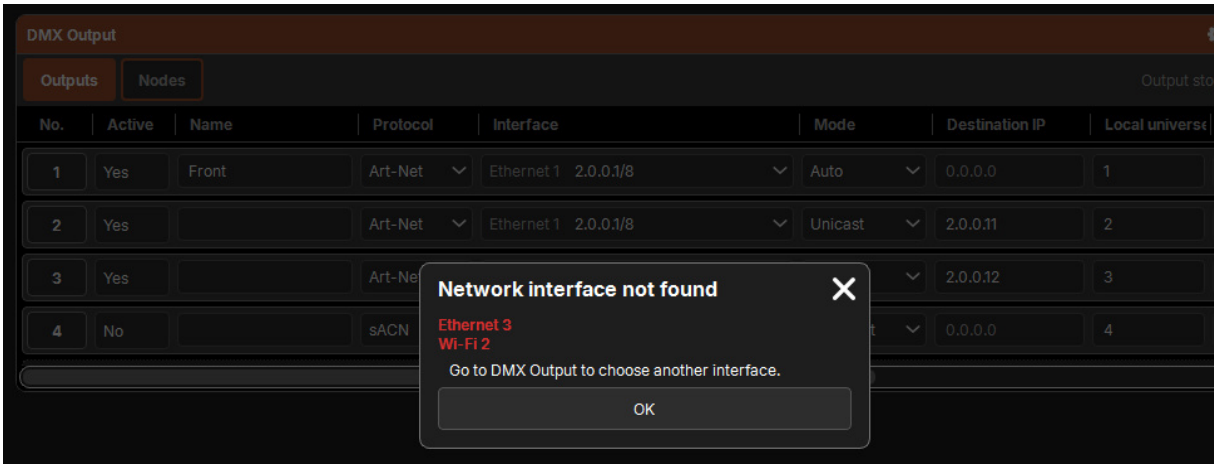
Changing the protocol resets the remote universe to the start of its numbering (0 in Art-Net, 1 in sACN).

Modes

Mode	What is sent
Auto (Art-Net)	each universe goes in **unicast** to the nodes that announce they output it; in **broadcast** if none is known (a node that does not answer ArtPolls must still receive) or if there are more than 40
Broadcast (Art-Net)	all universes to the whole network; very common in live events, simple, but loads every device on the network
Unicast	to the **Destination IP** only
Multicast (sACN)	each universe to its multicast group: only subscribed receivers get it

ArtSync (the synchronization of Art-Net universes) always goes in broadcast. An sACN universe that is no longer sent (output turned off, software closed) receives three stream-terminated packets: receivers release it at once.

Network interfaces



The **Interface** list only shows real cards, numbered by type with their address

("Ethernet 1 | 192.168.1.10/24", "Wi-Fi 1 | ..."), then the loopback. Virtual cards (Hyper-V, VirtualBox, VMware, VPN, Docker...) are left out. A card plugged in while running appears by itself.

If an active output targets a card missing from the computer (a show from another machine, a card removed):

- at startup and when the show is opened, the **Network interface not found** popup reports it and invites you to choose another card here;
- the row keeps its card, shown **not found**, and sends nothing while it is missing.

Attention: Windows must allow the software on the network (firewall, **private** networks). If nothing comes out, check this first, then that the chosen card is on the same network as the nodes.

Which universes are sent

Only universes where at least one fixture is patched are sent: there is no need to set up empty universes. The engine sends **44 frames per second**.

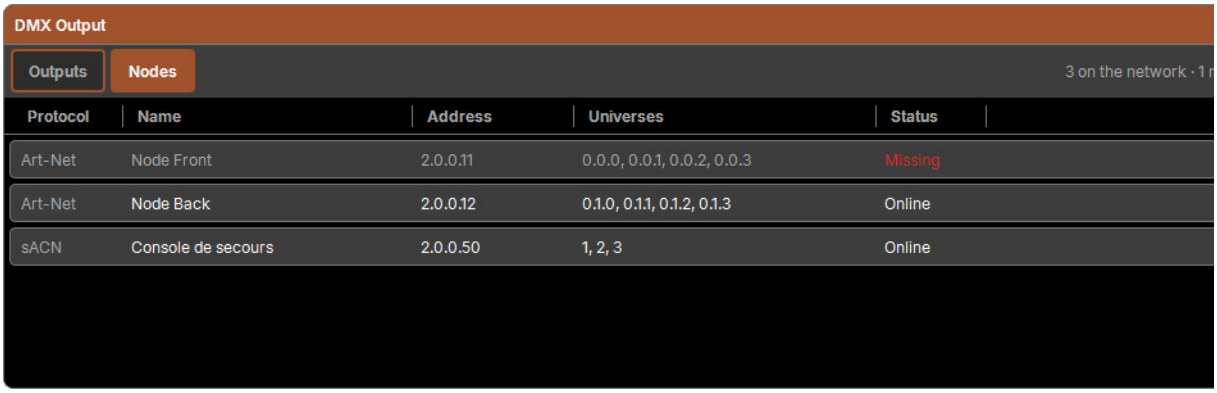
Selection and sequential entry

Cells are selected as everywhere (a rectangle, or a title for the whole column; see selection). Then a value typed in the command line and **Enter**:

- **Destination IP**: in sequence (`192.168.1.11` gives .11, .12, .13... in selection order);
- **Local universe** and **Remote universe**: in sequence, each output starting after the universes of the previous one;
- **Count**, **Priority**, **Delay**: the same value everywhere.

A choice (Active, Protocol, Interface, Mode, Preview) made in a selected cell applies to the whole selection.

Nodes



The **Nodes** category shows what the software sees on the network; nothing is set here.

- **Art-Net**: an ArtPoll every 2.75 s; each node that answers is listed with its name, its address and its universes (as Net.Sub-Net.Universe). With no answer for 10 s, it stays listed, marked **Missing**, and is no longer served by Auto mode.
- **sACN**: the sources on the network, discovered through Universe Discovery (every 10 s).

- Discovery **only runs on the interfaces of active outputs**, so as not to send anything on a network that is not yours. With no active output, the list stays empty, and the summary says so.
- If another program already uses the Art-Net port (6454), the summary says so: nodes cannot be detected.

What the engine does

At the end of the bar (Outputs category), the engine says what it is doing: "Frame: 0.4 ms · send 0.1 ms · 6 universes · 12 sends · 6 calculator(s)", and the skipped frames if any. **Output stopped**: the engine is not running; **No universe sent**: nothing is patched, or no output is active.

The calculation runs apart from the interface, on all processor cores but two: the screen can lag without the light stuttering.

Big shows

The software limits nothing: no limit on universes or fixtures. If the computer cannot keep up (frames are skipped, repeatedly), the command line says it once: "DMX Output: the computer cannot keep up, frames are skipped". Nothing is cut; look at the timings in this window, and lighten the load (fewer universes per output in broadcast, unicast or Auto rather than broadcast, a more powerful computer).

Not yet: RDM and merging several consoles.

All the keys

The keys of the command pad, one line each. An action key lights up and waits for its object; a second press turns it off.

Actions

Key	In short	Details
Store	the programmer into the object tapped next (cue, group, preset, view, Fader, Button)	The pad, Storing cues
Delete	deletes the tapped object, or the selection (**Enter**)	Lists
Copy	the source, then the target; **Enter**: at the end	Lists
Move	like Copy, the source leaves its place	Lists
On	turns on: a Cue List, a main window, an attribute in the programmer	The pad
Off	turns off: a Cue List, a main window; removes from the programmer	The pad
Off Off	quick double press: **Park**	Park
On On	quick double press: **Unpark**	Park
Assign	assigns a Cue List to a Fader or a Button, an object in a Layout, a group in a World	The pad
Release	the targeted attributes get a Release in the programmer	The pad
Update	updates the presets and running cues with the programmer	The pad

Objects

Key	In short	Details
Cue	a cue number of the Cue List shown: "Cue 5", "Cue 2 Thru 8"	The pad
Fixture	fixtures for the programmer: "Fixture 1 Thru 50", "101.3"	The pad
Fixture Fixture	double press, after Park or Unpark: "Selection"	Park

Numeric pad

Key	In short
0 … **9** **.**	a number, a value; in a field being edited, types into it
Thru	a range: "1 Thru 50" (grandMA's Thru)
⌫	erases the last character, otherwise the object, otherwise the action
Enter	executes the command line; in a field, confirms it

Programmer

Key	In short	Details
Clear	in three steps: releases the selection, deactivates the values, empties the programmer	The pad
Highlight	toggle: the selection at its Highlight values	The pad
Solo	toggle: the other fixtures at 0	The pad
< >	the selection, fixture by fixture	The pad
Set	the whole selection / the sub-selection; held with **< >**: 1 in 2, 1 in 3...	The pad

Undo

Key	In short	Details
Undo	cancels the command in progress, otherwise the last change to the show	Undo and Redo
Redo	redoes the last undone change	Undo and Redo
Esc	clears the cell selection; on the second press, cancels the action in progress	The pad

Keyboard shortcuts

E&S Lighting can be run entirely on screen, but a keyboard saves time: it feeds the command line like the command pad.

Everywhere

Key	Effect
F11	full screen, and back to the previous size
Esc	like **Esc**: clears the cell selection, then cancels the action in progress
Ctrl+Z	like **Undo**: cancels the command in progress, otherwise the last change to the show
Ctrl+Y (or **Ctrl+Shift+Z**)	like **Redo**

The command line

When no field is being edited, these keys go to the command line:

Key	Effect
digits	the number (after **Cue** or **Fixture**), or a value for the selection
. or **,**	the point of a number ("1.5", "101.3"); a comma counts as a point
-	a negative value (an offset)
à	a range, like **Thru** (on a French keyboard, the "à" key without Shift)
Enter (or the numeric keypad's Enter)	like **Enter**: executes the line
Backspace	like ⌫: erases the last character, otherwise the object, otherwise the action

Digits work from the top row (with Shift on an AZERTY keyboard) as well as from the **numeric keypad**. With Ctrl or Alt, they are shortcuts: nothing is typed in the command line.

Actions (**Store**, **Delete**...) and objects (**Cue**, **Fixture**) have no keyboard shortcut: they are chosen on the command pad.

In a field

A field being edited (a name, a value) keeps its keys:

Key	Effect
Enter	confirms
Ctrl+Z / Ctrl+Y	undo / redo the typing in the field
Esc (name of a tile)	restores the previous text

The command pad also types into the field being edited: digits, point, ⌫ and **Enter**.

Lists

Gesture	Effect
mouse wheel	scrolls the list; does not change a closed drop-down list
Esc (drop-down list open)	closes it without changing anything

Frequently asked questions

No light comes out

Check, in this order:

1. **The license**: without an active license, the DMX output is cut (a gray veil on the Layouts reminds you). "Add a window" → Setup → **License** tells you why. See License.
2. **An output is set up**: **DMX Output** window, at least one line with **Active** on Yes, with the right network **interface**, and universes that cover those of your patch. In Unicast, a **Destination IP** of 0.0.0.0 makes the line inactive. See DMX Output.
3. **The patch**: a universe without any patched fixture is not sent. The status line of the DMX Output window says "No universe sent" in that case.
4. **The Windows firewall**: on first launch, it asks you to allow the software; if it was refused, no frame goes out. Allow **E&S Lighting** for private networks in the firewall settings.
5. **The node**: it must be on the same network as the chosen interface (same IP address range), and listen to the universes sent (in Art-Net, watch out for the Net.Sub-Net.Universe numbering).
6. **The fixtures themselves**: a parked fixture stays frozen (see Park); **Solo** on sets all unselected fixtures to 0.

"Network interface not found" at startup

A DMX output of the show uses a network card that is not on this computer (show coming from another computer, card unplugged, USB adapter). Open **DMX Output** and choose an interface that is present. The line keeps the old interface, shown as "not found", until you change it.

During a show, a card that disappears is reported in the command line ("Network card not found..."): its outputs are no longer sent until it comes back.

"The machine can't keep up, frames are dropped"

The DMX output calculation can no longer keep up the pace (44 frames per second): too many universes or effects for this computer. The software cuts nothing, it warns you. The **DMX Output** window shows the calculation time of a frame and the dropped frames. Close other programs, or move to a computer with more cores.

The software won't open a second time

That's intended: only one E&S Lighting at a time per computer (there could only be one DMX output). If it doesn't open at all, it may already be running, hidden: look at the taskbar, or close it in the Task Manager.

I made a mistake

Undo (or Ctrl+Z) cancels the last change; the **History** window lets you go back several steps at once. See Undo and Redo.

Where is my show saved? How do I back it up?

Every change is written immediately to `Documents\E&S`

`Lighting\Shows\<name>.esl`: there is nothing to save before closing. To back it up, copy this file, or use **Copy** on the show in the **Shows** window. See Shows.

My windows have disappeared

- A main window that is turned off: **Windows** window, tap its button, it turns back on.
- The windows of a screen: recall a **View** (see Windows and Views), or reopen them through "Add a window" (a tap on an empty spot).
- **Clear the screen** (in "Add a window") closes all the windows of the screen: **Undo** does not bring them back (it is not a change to the show), but a View does.

Touchscreen

Everything is done with your finger: a tap for a button, one finger dragging to select (rectangle), two fingers to scroll a list. The command keypad keys type into the field being edited: no keyboard needed. See Selecting.

Small screen

The interface is designed to work from **1024 × 600**. On a small screen, use **Views** to switch from one layout to another rather than stacking windows, and **F11** for full screen.

A fixture is not in the library

Look for it on **GDTF Share** from the Profiles window, import a `.gdtf` file from the manufacturer, or create its profile yourself. See Profiles and GDTF.

The software is in English (or in French)

"Add a window" → Setup → **Language**, then restart the software. By default, it uses the language of Windows.

Does it work without internet?

Yes, for 30 days after the last license check. See License.

I found a bug / I have an idea

The **Report a bug / an improvement** button, at the top of the window: describe what you were doing, attach the log and the screenshot. You follow up on the website, **My reports** page. See License.