

LightSync — what you get and what you can do with it

A document for lighting technicians. It explains what each file coming out of LightSync contains, how it connects to a real rig, and why the program is built this way.

First things first, to save you time

LightSync is not a console. It does not replace your desk, it stores no cues, and it never touches your patch.

It is a **light-show composer that works by listening to the music**, and it sends the result over Art-Net to whatever you have connected. It analyses the track — tempo, bar grid, sections, energy, hits — and a director decides what the fixtures do at every moment.

If your job is programming a locked, timecoded show, this will not do it for you. If your job is making the lights respond to whatever is playing — DJ sets, bands with a changing setlist, events where nobody knows what will be playing in twenty minutes — that is where it lives.

The two files, and why there are two

This is the distinction to understand before anything else.

The stage (.mvr) — describes the rig

It contains the fixture models, which DMX mode each one is in, where they hang in space and at which addresses. It is the standard My Virtual Rig format, so Capture, grandMA3, Vectorworks, BlenderDMX and any serious tool in the trade can open it.

Each model's GDTF file travels inside it — that is how the format works: whoever receives it sees the real fixtures, not markers.

This file describes one specific rig. If yours is different, it helps you understand the design — it will not run it for you.

The show (.json) — describes the intention

It contains the style, the choreography, the per-track presets, the setlist and every director setting.

This file does not mention a single fixture.

Why that separation matters

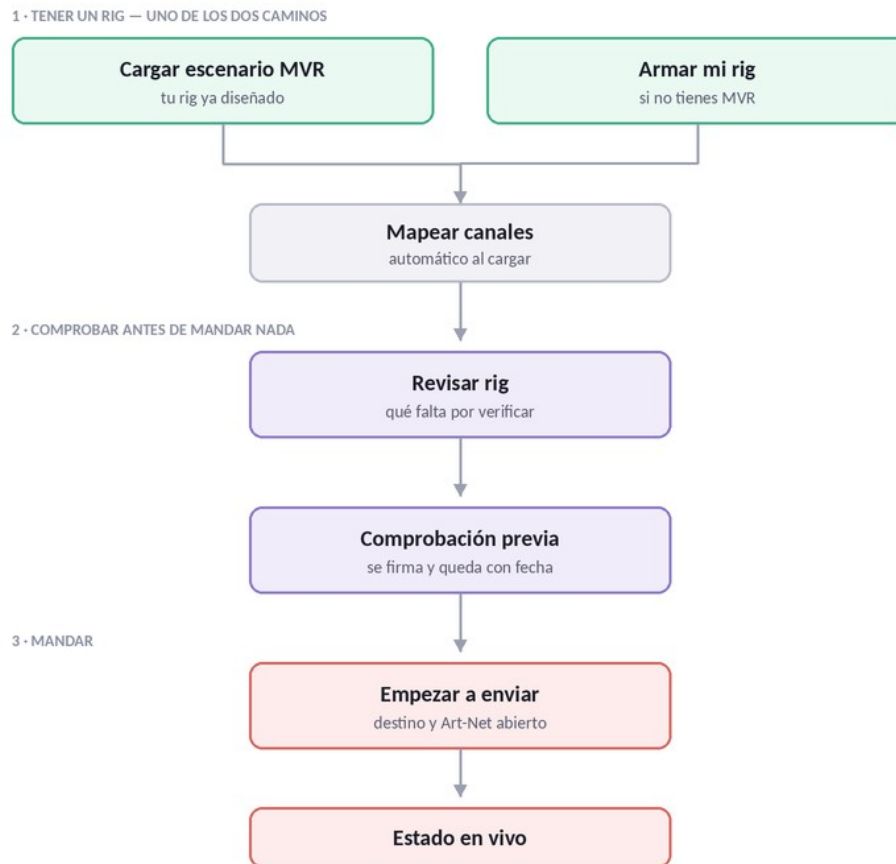
Because it means **the show travels from one rig to another and still makes sense**.

The director does not program fixtures: it composes with **music and with zones**. It distributes across proportional thirds of whatever rig it finds, not by fixture IDs. If the original design had 146 fixtures and you have 40, the same show spreads across yours.

A timecoded show is tied to the fixtures it was programmed with: change the rig and you reprogram. Not here.

How you connect it to what you have

The opposite of what it looks like: **you load your rig, and you put the show on top of it.**



The sequence is always the same and the program shows it on screen, marking which step you are on. Having a rig admits two alternative paths; from there on, everything is a single line.

Path 1 — you have your rig as an MVR

Drag it in and that is it. LightSync reads each fixture's channels from its .gdxf and its addresses from the patch. Nothing is declared by hand.

This is the good path if you come from Vectorworks, from a console that exports MVR, or from a visualiser.

Path 2 — no MVR, but you know what you have

There is a template where you declare your inventory: **how many, of which model, and in which part of the venue.** Out of that comes a stage — placed, grouped and patched.

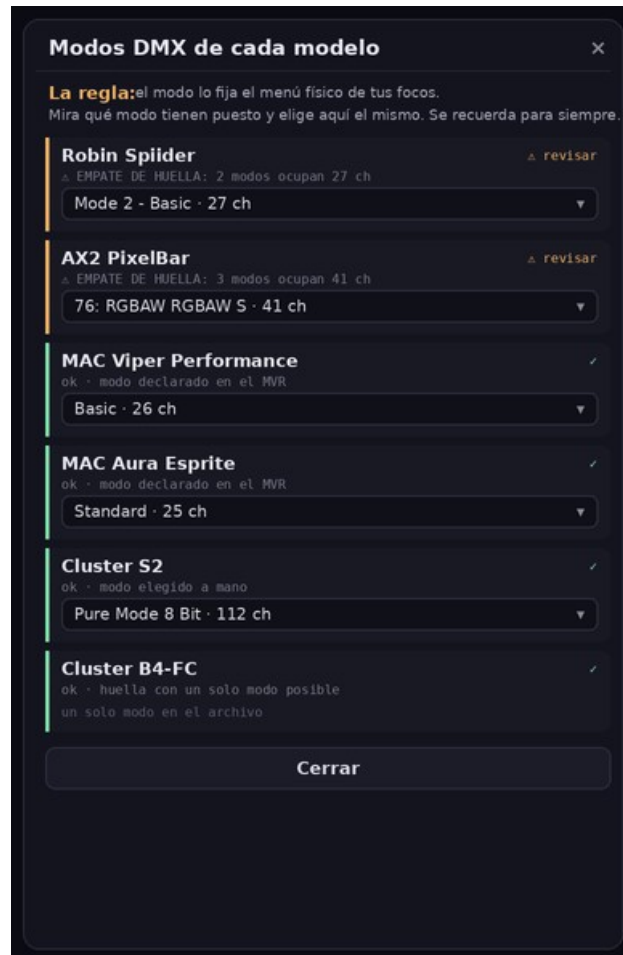
That third column — **where** — is not decoration. If you declare eight moving heads and the program places them on the front truss when in your venue they sit on the floor, a sweep that goes left-to-right on screen will go top-to-bottom on your rig. No error would appear: the show would simply look wrong.

You can drag your own .gdxf files onto that list so the patch comes out with each model's real footprint.

Path 3 — you just want to see it running

There is a demo stage with generic fixtures. It is no good for sending to real fixtures — its channels are ours, not any particular model's — but it shows you what this is about in two minutes.

The DMX mode is chosen on screen



When a fixture admits several modes with the same channel footprint, no program can guess which one you have set: the fixture's physical menu decides that. The **Modes** panel (next to Check rig) lists every model with its status — verified or to review — and a dropdown with its possible modes. You check your fixture's menu, pick the same one here, and the choice is saved per model: when the stage reloads, it reapplies itself.

None of this blocks the stage. The 3D and the show always run; modes only decide which byte lands on which channel when you send Art-Net to real fixtures. An unresolved mode is an amber notice, not a barrier.

What LightSync does with your GDTFs

This part matters to anyone about to send Art-Net to real fixtures, because it is where the patch either lines up or does not.

It reads the channels from the file itself

There is no internal model table. Each fixture's channel map comes from **its GDTF**, written by the manufacturer. Dimmer, colour, shutter, pan, tilt.

It works out which mode is set — and says so when it cannot

A model can have many DMX modes, and the same fixture occupies a different number of channels in each. LightSync resolves it in this order:

- **If the MVR declares the mode** and that mode exists in the GDTF, it is used. That is the datum, not a deduction.
- **If not**, it measures the spacing between the addresses of that model's fixtures. With several fixtures patched in a row, that distance gives the mode away.
- **If that measurement fits more than one mode**, it says so. Matching one mode is evidence; matching three is a tie.
- **If there is only one fixture of that model**, there is no spacing to measure, and it says plainly that the mode is unverified.

In the last two cases the mode can be set by hand. The point: **the program never assumes what it could not verify.**

It understands multi-cell fixtures

Many fixtures are several lights in one: a pixel bar, a head with independent rings, a cluster of tubes. Their channels are declared in two different ways in GDTF — repeating the attributes, or describing one block and repeating it by geometry — and LightSync handles both.

In practice: if your bar has eight pixels, all eight light up, not just the first one.

It parks what it does not drive

Channels the show does not use — colour temperature correction, frost, iris, zoom, focus — are set to the **rest value the manufacturer declares**, not to zero. The fixture comes up with the face its maker considers neutral.

And the shutter and colour wheel are opened by reading from the file itself **which value means open**, instead of assuming it is 255.

Colour, adding and subtracting

An LED par mixes additively, with red, green and blue emitters. A discharge head mixes subtractively, with cyan, magenta and yellow flags in front of a white lamp. Opposite scales. LightSync reads from the GDTF which one each fixture is and sends accordingly.

The Art-Net output

It comes **from the program itself**, with nothing extra to install and no scripts to run.

- It writes whole universes and never spills into the next one.
- A fixture is never split across two universes when auto-patching.
- On stop, zeros are sent.
- If one model has a corrupt profile, it costs that fixture, not the show: the rest keep receiving.

Mind the numbering: Art-Net universe 0 is Universe 1 on most consoles and visualisers. Nobody's fault — it is how the standard is written.

Which channels are written

Intensity, colour, shutter, and **pan and tilt**. Movement is sent **relative to each fixture's rest position**: how it hangs and where it is pre-aimed remains your decision, and the show moves from there. Fixtures that do not move receive no movement.

What it does NOT do

Said upfront, which is more useful than finding out mid-load-in:

- **It does not program cues or scenes.** There is no sequence list to execute.
- **It does not replace your console** on a locked, timecoded show.
- **It does not invent fixtures.** If the original show leaned on moving heads and you have pars, it will look different. The director adapts to what is there.
- **It does not touch your patch.** It reads the one you give it.
- **It does not know your fixtures by heart.** It needs the GDTF, which is free on GDTF Share.

Previewing

The stage renders in 3D inside the program, with beams and haze, reacting to the music in real time. To judge a show, that is enough.

If you want it in your usual visualiser, export the stage as MVR, open it in Capture, grandMA3 or BlenderDMX, and send the Art-Net there. Bear in mind that a big rig with volumetric beams is heavy load for any general-purpose 3D engine.

And one thing about the file you receive

If someone sends you an MVR that came out of LightSync, that file **states which DMX mode every fixture is in**. Not every program writes it — many drop in generic filler and leave you guessing — and that turns a stage into half the information.

Here it comes written. Load it back into LightSync and it resolves itself: no ties, no deductions, no warnings.