# DreamShow — user manual

Version 2026.287

## Contents


**Getting started**

- Welcome to DreamShow
- First steps
- Creating your first show (music or theatre)
- Declaring your fixtures (venue wizard)
- Installation, languages and updates
- DreamShow is free

**The main screen**

- Main screen overview
- The Scenes panel
- Scene settings
- Selecting fixtures and adjusting the light
- Palettes
- Libraries
- Log, Audio and Clock
- 2D and 3D simulation

**The timeline**

- The scene timeline

- Light and effect blocks
- Fades and mixes
- Sound and video on the timeline
- Waits (key, time, date, sun)
- Wait for a spoken line (voice sync)
- Jumps and scene labels
- Parallel sequences
- Architectural lighting
- Dynamic effects
- Pixel mapping

**Aiming, zones and trajectories**

- Aiming and covering a zone with moving heads
- Trajectories
- Moving-head calibration

**The venue**

- Fixtures, addresses, positions
- The fixture library
- DMX connections
- Manual controls
- Video screens
- Switching, renaming, duplicating a venue

**DMX and control**

- DMX values and DMX faders
- Visualiser mode
- Sound-to-light and tempo
- Timecode (LTC / MTC)
- MIDI controllers
- The web remote
- The OSC remote
- The remote (GO keys)

**The performance**

- Show mode and the cue window
- Recovering after a crash or a power cut

**Creating with an AI**

- Creating a show with an AI
- DMX stream by an AI
- AI assistant (set the scene)
- Connect an AI (MCP)

**Stage devices**

- The standalone player
- The follow spot on a tablet
- Backstage displays
- The teleprompter
- Humanoid robots
- LED masks
- Connected sockets

**Files**

- Shows, files and projects

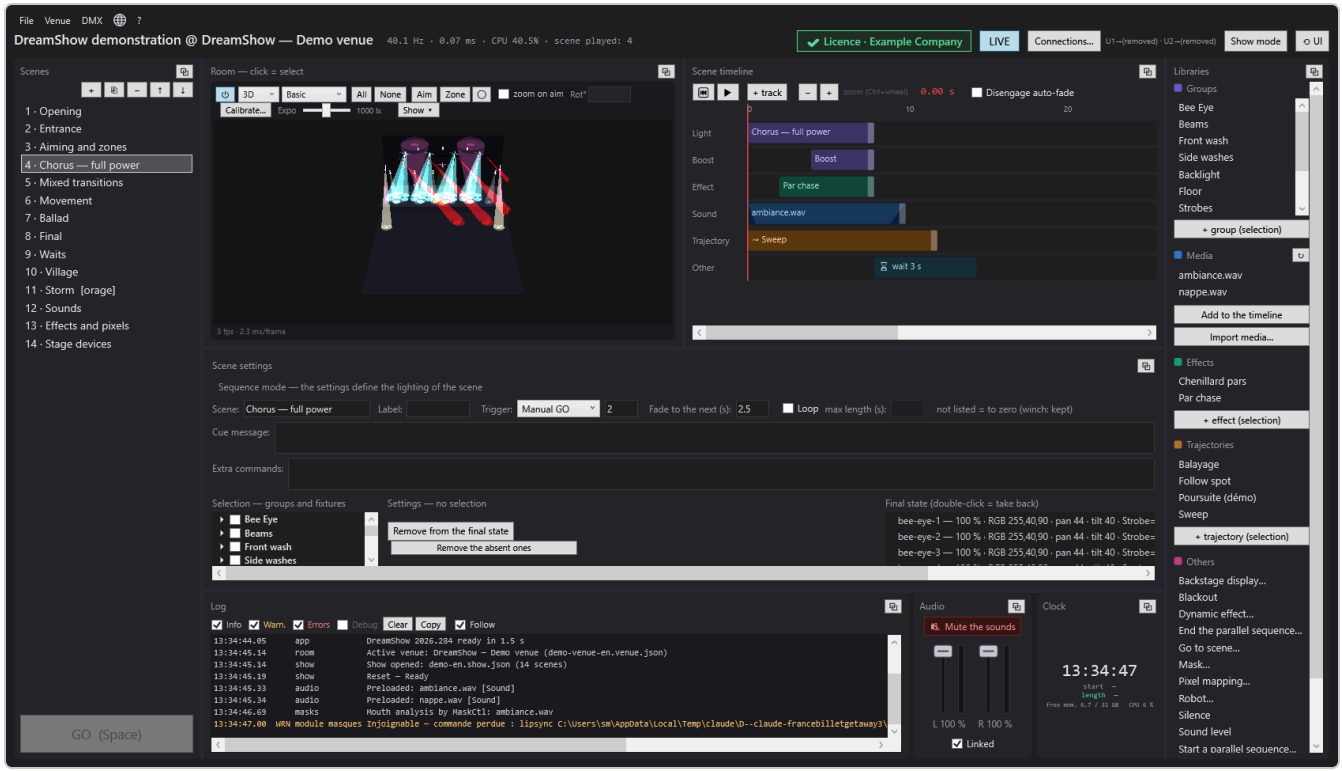
**Help**

- Troubleshooting
- Keyboard shortcuts
- Glossary

# Welcome to DreamShow

DreamShow is lighting, sound and video show control software for Windows. It drives your fixtures over DMX, plays sound and video, and coordinates other stage devices. Everything is prepared as a series of scenes, each with its own timeline, and then played back during the performance, scene after scene, with the **GO** button.

This manual explains every window, every setting and every button. It is written for operators, venue technicians, DJs and installers who are discovering DreamShow.




## What DreamShow does

- **Light**: your fixtures are declared in a **venue**, with their DMX address and their position in metres. You set colours, intensities, moving-head positions (motorised heads), effects and trajectories. DreamShow sends the result as DMX over the network (Art-Net, sACN) or through a USB interface.
- **Sound and video**: sound files are played by DreamShow; videos are shown on screens (Windows monitors or video players on the network).
- **The cue list**: a show is a series of **scenes**. Each scene has a **timeline** (tracks and blocks: light, effects, sound, video, waits…) and a **final state**. During the performance, the cue window (Show mode) runs the scenes on **GO**, automatically or at a set time.

- **2D and 3D simulation**: the venue is displayed with its beams, synchronised with the timeline. You can prepare everything without a single fixture connected.
- **Stage devices**: standalone player (plays a show back without a PC), follow-spot tablets, backstage displays, humanoid robots and LED masks.

DreamShow is free: the whole software is open, including control of real equipment (DMX output, IP video players, standalone player, tablets, displays, robots, masks). See DreamShow is free.

## How to use this help

- **F1** opens the help page that matches the current window.
- **Search** finds a word across all the pages.
- **Links** in the text lead to another page of the manual. Every page ends with a **See also** section.

### Conventions used in this manual

- On-screen labels are written in bold, exactly as in the software: **Save as...**.
- A menu path is written with the `>` sign: **File > Open a show...** means "open the File menu, then click Open a show...".
- DreamShow's help menu is called **?**; the language menu is shown as a globe (its tooltip reads **Langue / Language**).

## Where to start

1. If you have just installed DreamShow, the **Welcome to DreamShow** screen opens by itself: watch the demonstration, create a first show on a piece of music, or declare your fixtures. See First steps.
2. To write a first show in two minutes: Creating your first show (music or theatre).
3. To describe your real equipment: Declaring your fixtures (venue wizard).
4. To understand the screen: Main screen overview.

## The main parts of the manual

### Getting started

- First steps
- Creating your first show (music or theatre)
- Declaring your fixtures (venue wizard)
- Installation, languages and updates
- DreamShow is free

### The main screen

- Main screen overview
- The Scenes panel
- Scene settings
- Selecting fixtures and adjusting the light
- Palettes
- Libraries
- Log, Audio and Clock
- 2D and 3D simulation

### The timeline

- The scene timeline
- Light and effect blocks
- Fades and mixes
- Sound and video on the timeline
- Waits (key, time, date, sun)
- Jumps and scene labels
- Parallel sequences
- Architectural lighting
- Dynamic effects
- Pixel mapping

### Aiming, zones and trajectories

- Aiming and covering a zone with moving heads
- Trajectories
- Moving-head calibration

### The venue

- Fixtures, addresses, positions
- The fixture library
- DMX connections
- Manual controls
- Video screens
- Switching, renaming, duplicating a venue

### DMX and control

- DMX values and DMX faders
- Visualiser mode
- Sound-to-light and tempo
- Timecode (LTC / MTC)

- MIDI controllers
- The web remote
- The OSC remote
- The remote (GO keys)

### The performance

- Show mode and the cue window
- Recovering after a crash or power cut

### Creating with an AI

- Creating a show with an AI
- DMX stream by an AI

### Stage devices

- The standalone player
- Follow spot on a tablet
- Backstage displays
- Humanoid robots
- LED masks

### Files

- Shows, files and projects

### Help

- Troubleshooting
- Keyboard shortcuts
- Glossary

## Good to know

- The installed version is shown in the title bar (for example “DreamShow 2026.282”) and in **? > About DreamShow...**. Always quote it when you ask for help.

## See also

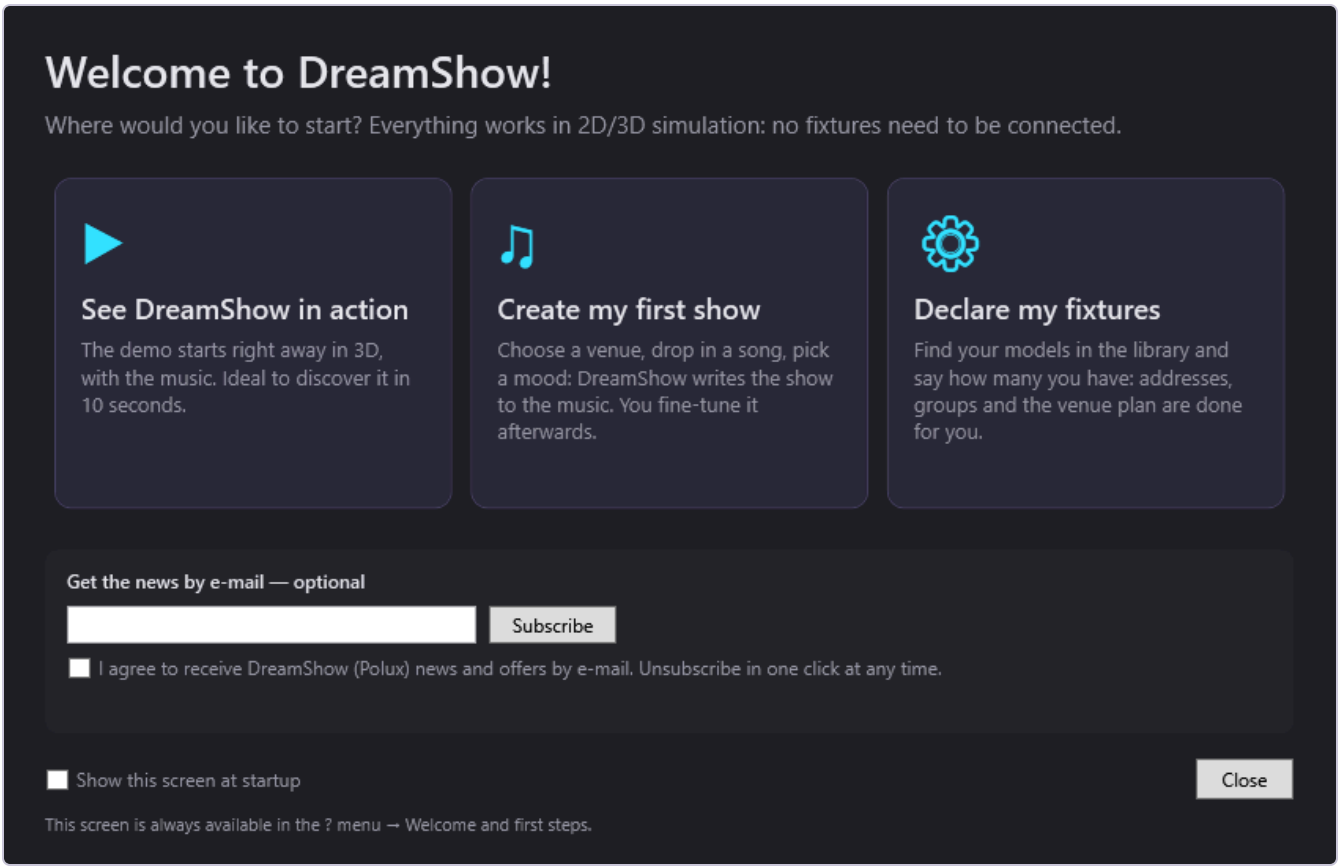
- First steps
- Main screen overview
- Glossary

# First steps

The first time you start it, DreamShow does not leave you facing empty panels: it offers three ways to begin, then a short list of six things to do once to learn the essentials. None of this is compulsory.

**To open it:** the welcome screen opens by itself after a fresh installation. After that: menu **? > Welcome and first steps…** for the welcome screen, **? > First steps checklist…** for the list of six things to do.

## The “Welcome to DreamShow” screen




At the top, the title **Welcome to DreamShow!** and the sentence: “Where would you like to start? Everything works in 2D/3D simulation: no fixtures need to be connected.”

### The three choices

Three large tiles. Clicking a tile closes the screen and starts that choice.

- ▶ **See DreamShow in action**: the demonstration starts straight away in 3D, with its music. DreamShow opens the demonstration show in the interface language (in English: “DreamShow

demonstration", on the venue "DreamShow — Demo venue"), switches the simulation to **3D**, selects the first scene and plays it from the start. If the current show has unsaved changes, DreamShow first asks you **Save the changes to the show?**.

- ♫ **Create my first show**: opens the **Create my first show** wizard. You choose the type of performance, a venue, one or more songs and a mood (or, for theatre, a list of moments); DreamShow writes the show. Everything is described in Creating your first show (music or theatre).
- ⚙ **Declare my fixtures**: opens the **Declare my fixtures** wizard. You look up your models in the library and say how many you have; DreamShow creates the venue with the addresses, the groups and the plan. See Declaring your fixtures (venue wizard).

### Receiving news by e-mail — optional

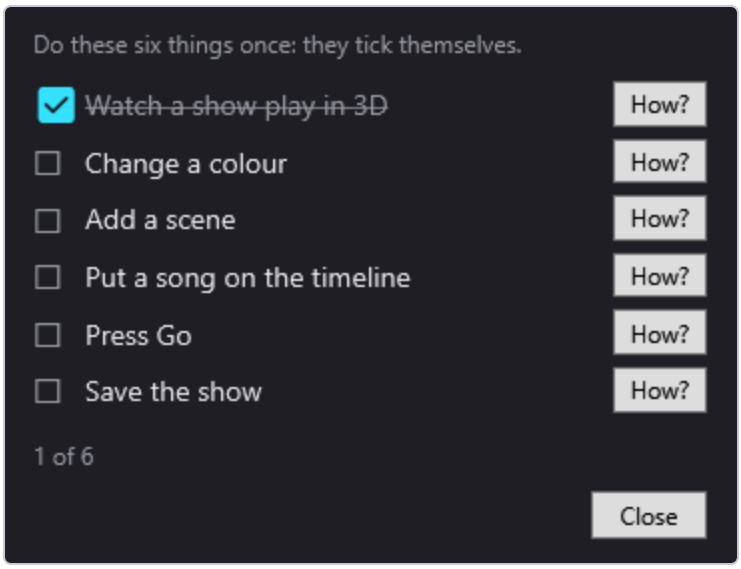
A frame offers to send you DreamShow news by e-mail. It is entirely optional and has no effect at all on the software.

- **Address field**: type your e-mail address.
- **I agree to receive DreamShow (Polux) news and offers by e-mail. Unsubscribe in one click at any time.**: check box, required to subscribe.
- **Subscribe**: sends the subscription. A message appears below the frame:
  - "Please enter a valid e-mail address." if the address is not in the form name@domain.ext;
  - "Tick the box to give your consent." if the box is not ticked;
  - "Sending…" while sending, then the service's reply;
  - "Service unavailable: please try again later." if the PC has no Internet access or the service does not respond.

### The bottom of the screen

- **Show this screen at startup**: when ticked, the welcome screen opens every time DreamShow starts. When unticked, it no longer opens by itself. After a fresh installation (no show opened yet on this PC), the box is ticked; as long as you have never changed it, an update does not bring the screen back on a PC that is already in use. The choice is saved when you close the screen.
- **Close**: closes the screen without starting anything.
- The last line reminds you that "This screen is always available in the ? menu → Welcome and first steps."

## The first steps checklist




A small window titled **First steps**, which stays open next to the main screen (at the bottom right) while you work. It says: “Do these six things once: they tick themselves.”

### The six things to do

Each line has a box (☐, then ☑ when done, and the text is then struck through) and a **How?** button.

| Thing to do | What ticks it |
|---|---|
| **Watch a show play in 3D** | the simulation is in **3D**, playback is running and the scene being played has gone past 2 seconds |
| **Change a colour** | a click in the colour palette, or a movement of the R, G or B sliders, while fixtures are selected |
| **Add a scene** | the **+** button above the scene list (duplicating a scene also counts) |
| **Put a song on the timeline** | a media file (sound or video) added to the timeline, with the **Add to the timeline** button or by dragging the file |
| **Press Go** | the **GO (Space)** button or the space bar in the main screen |
| **Save the show** | **File > Save** |

A ticked item stays ticked: the list is saved in DreamShow's settings.

### The How? button

It opens a yellow bubble attached to the control concerned, with the explanation:

- **Watch a show play in 3D**: choose “3D” in the list at the top of the venue view, then press Go (space bar). The menu **? > Welcome and first steps…**, tile **See DreamShow in action**, does it for you.

- **Change a colour**: click a scene on the left, tick a group in the selection (at the bottom), then click in the colour palette or move R, G, B.
- **Add a scene**: the **+** button above the scene list, on the left. Each scene is a moment of the show, triggered by Go.
- **Put a song on the timeline**: in **Media** (on the right), **Import media…** then **Add to the timeline**; or drag the file straight onto the timeline.
- **Press Go**: the big Go button at the bottom left, or the space bar: you move on to the next scene, just as during the show.
- **Save the show**: **File > Save**, or Ctrl+S.

The bubble disappears after 12 seconds, or as soon as you click elsewhere. If the target control is not visible (hidden panel), the bubble appears to the left of the **First steps** window.

### The bottom of the list

- A counter shows how far you have got, for example “4 of 6”.
- When all six things are done, it is replaced by: “Well done! You know the essentials. The rest (effects, trajectories, video, robots…) is shown in the videos and the brochure on the DreamShow website.”
- **Close**: closes the list. It can be reopened with **? > First steps checklist…**.

After a fresh installation, the list opens by itself after the welcome screen, as long as the six things have not all been done.

## Empty panels guide you

Two hints appear on their own when there is nothing to see yet:

- **In the timeline**, when the selected scene is empty (no block, no setting in its final state) and Show mode is not open: “Empty scene. To light it: tick fixtures in “Selection” (at the bottom) and set intensity and colour. To animate it: drag a song, an effect or a trajectory from the library (on the right) onto the timeline.”
- **In the simulation**, when the active venue has no fixture: “This venue has no fixture yet.”, with a **Declare my fixtures…** button that opens the venue wizard.

These hints disappear as soon as the scene or the venue is no longer empty.

## Step by step

### Discovering DreamShow in one minute

1. Open **? > Welcome and first steps…** (or wait for the welcome screen the first time you start it).

2. Click ▶ **See DreamShow in action**.
3. The demonstration plays in 3D with its music. The item **Watch a show play in 3D** ticks itself after two seconds.
4. Press the space bar to move on to the next scene (item **Press Go**).

### Doing the six things

1. Open **? > First steps checklist…**.
2. For each unticked line, click **How?** and follow the bubble.
3. Finish with **File > Save**.

### No longer showing the welcome screen at startup

1. Open **? > Welcome and first steps…**.
2. Untick **Show this screen at startup**.
3. Click **Close**.

## Good to know

- After a crash, the recovery offer comes before the welcome screen. See Recovering after a crash or power cut.
- The item **Save the show** is only ticked through the **File > Save** menu: the Ctrl+S shortcut does save the show, but does not tick the line.
- The **How?** bubble for the item **Save the show** mentions an “automatic backup”: DreamShow keeps a copy of the previous version each time you save ( `autosave` folder), but does not save the show by itself. See Shows, files and projects.

## See also

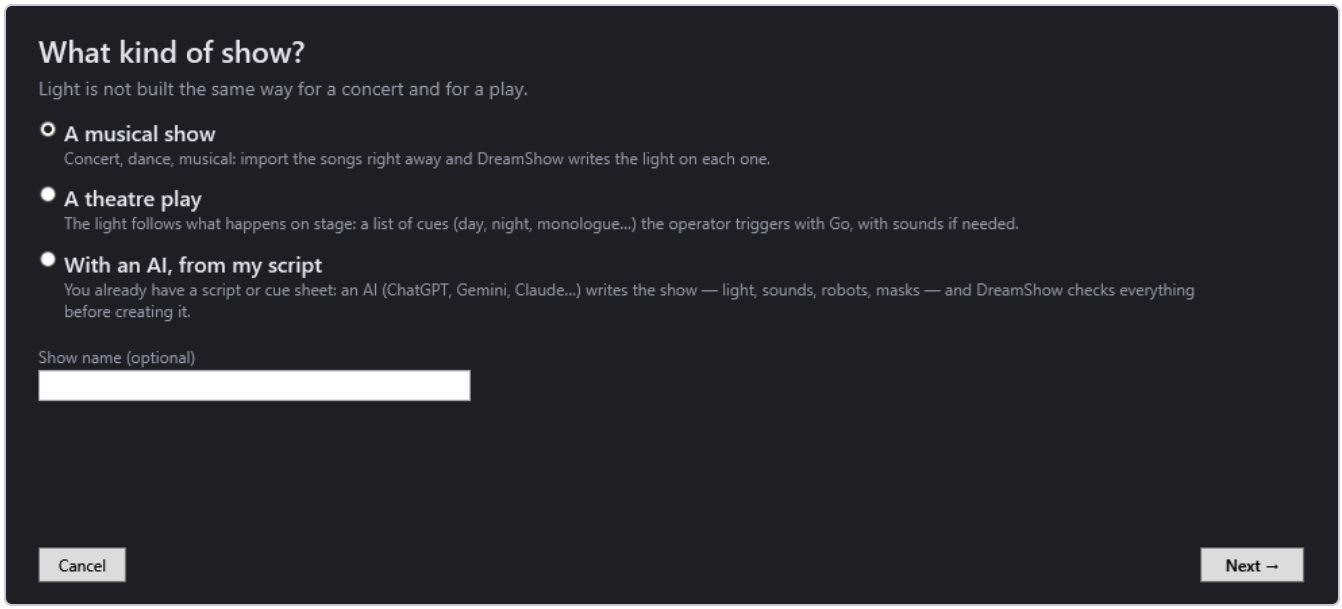
- Creating your first show (music or theatre)
- Declaring your fixtures (venue wizard)
- Main screen overview
- 2D and 3D simulation
- The Scenes panel

# Creating your first show (music or theatre)

The **Create my first show** wizard writes a complete show for you that works straight away: on one or more songs for a musical show, or from a list of cues for a play. You start from something that works, then you fine-tune it in the timeline.

**To open it:** menu **? > Welcome and first steps…**, tile **♫ Create my first show**. It is also offered at the end of the **Declare my fixtures** wizard (see Declaring your fixtures (venue wizard)).

If the open show has unsaved changes, DreamShow first asks **Save the changes to the show?**.




## Moving through the wizard

- **Next** →: goes to the next page. On the last settings page, it becomes **Create the show** ✨.
- ← **Back**: returns to the previous page (not shown on the first page or on the creation page).
- **Cancel**: closes the wizard without creating anything. The window cannot be closed while the show is being created.
- Each page title is preceded by its number, for example “2/5 · Where does the show take place?”. The musical show has 5 pages, the play 4.

## Page 1: What kind of show?

“Light is not built the same way for a concert and for a play.” Choose one of the three options:

- **A musical show**: concert, dance, musical. You import the songs straight away and DreamShow writes the light on each one. This is the default choice.
- **A theatre play**: the light follows what happens on stage. You start from a list of cues (day, night, monologue…) that the operator triggers with Go, with sounds if needed.
- **With an AI, from my script**: you already have a script or a cue sheet. A conversational AI writes the show (light, sounds, robots, masks) and DreamShow checks everything before creating it. With this choice, **Next** → closes the wizard and opens the **Create a show with an AI** window: see Creating a show with an AI.

- **Show name (optional)**: the name of the show. If left empty, DreamShow uses:
  - for a play: “My play”;
  - for a musical show with a single song: the song's file name (without its extension);
  - for several songs: “My show”.

## Page 2: Where does the show take place?

“Take a ready-made venue to try: you can declare your real fixtures later.” Choose a venue:

- **Small stage**: 4 LED PARs in front, 2 moving heads upstage, 2 floor lights on the back wall. A 10 × 14 m room, with a stage 4 m deep. Groups: **Front**, **Moving heads**, **Back wall**.
- **Bar, club or party hall**: 6 LED PARs, 4 moving heads, 4 cell washes and 2 strobes. A 14 × 18 m room, with a 5 m stage. Groups: **Front**, **Moving heads**, **Cells**, **Strobes**.
- **Small theatre (no moving heads)**: the most common theatre rig: 6 conventional fixtures (PCs on dimmers) in front, 4 LED PARs as backlight, 3 LED floods on the cyclorama. No moving heads. A 10 × 14 m room, with a 5 m stage. Groups: **Front**, **Backlight**, **Cyclorama**.
- **My current venue: …** (followed by the name of the active venue): the fixtures already declared in DreamShow. This option only exists if a venue is loaded.

The choice selected in advance depends on the type: **Small stage** for a musical show, **Small theatre (no moving heads)** for a play, **My current venue** when you come from the venue wizard. A different choice that you made yourself is kept.

A ready-made venue is created only once in your data folder: if it already exists (same name), DreamShow reuses it as it is, without duplicating or overwriting it. Its fixtures are given consecutive DMX addresses, starting from address 1 of universe 1.

## Musical show

### Page 3: Which songs?

“Add one or more songs from your computer (mp3, wav, m4a, wma…), in show order. You can also drag them onto this window.”

- **Add songs…**: opens the file picker (several files at once). Accepted formats: mp3, wav, m4a, aac, wma, flac. You can also drag files onto the window; files in any other format are ignored.
- **The list**: the songs numbered in show order ("1. ♫ file name").
- **Remove**: removes the selected song from the list (the file itself is not touched).
- ↑ and ↓: move the selected song up or down one place.
- **Play the songs one after the other by themselves (otherwise Go between songs)**: only appears from two songs upwards. When ticked, each song starts as soon as the previous one ends. When unticked (the default), a Go is needed between songs.

If you click **Next** → without any song, the message "Add at least one song." is displayed.

### Page 4: Which mood?

"DreamShow follows the music: quiet passages stay soft, choruses build up."

- **Calm**: soft colours, slow fades, wide movements. Colour changes take 2 seconds, effects run half as fast.
- **Energetic** (the default): bold colours, sharp changes, pulses on every beat. Changes take 0.4 seconds, effects run twice as fast.
- **Party**: rainbow, rotating colours, party mood. Changes take 1 second.

Click **Create the show** ✨.

### What DreamShow writes for a musical show

DreamShow listens to each song ("Listening to the song: tempo, beats, verses and choruses…"): it extracts the tempo, the beats, the downbeats of the bars and the main sections. It then cuts each section into musical phrases of 8 bars, and gives each phrase an energy level (calm, medium, strong) by comparing it with the other phrases of the same song: so every song has its soft passages and its strong passages.

**One scene per song**, named after the song, triggered by Go, with the cue note "Go starts the music and the light." Its timeline contains up to four tracks:

- **Sound**: the song, from start to finish.
- **Light**: one light block per phrase, which fades into the next one. Two colours from the mood are chosen for each phrase: every other colour group takes the first, the others the second; moving-head groups take the colours in the reverse order. The intensity depends on the energy of the phrase: 55 % (calm), 80 % (medium), 100 % (strong), a little more for moving heads. The moving-head position changes slightly from one phrase to the next, and also with the energy (strong phrases have their own tilt). The first phrase fades in (over 1 to 3 seconds, up to the first beat) and the last one fades out over 3 seconds at the end of the music.
- **Movement** (if there are moving heads): one dynamic effect per phrase, locked to the tempo and started on a beat: a small vertical swing on calm phrases, circles on medium phrases,

figures of eight on strong phrases.
- **Pulse** (if there are colour fixtures): one intensity dynamic effect per phrase: slow breathing on calm phrases (except with the **Energetic** mood, where they stay still), breathing over two beats on medium phrases, and on strong phrases sharp pulses on every beat (**Energetic**) or every two beats (**Calm**), or a rotating rainbow (**Party**).

The colours: deep blue, amber orange, violet, sky blue and peach with the **Calm** mood; red, cyan, magenta, white, orange and blue with **Energetic**; every hue of the rainbow with **Party**. Each song starts at a different place in the palette, so that no two songs look alike.

**A last "Bows" scene**, triggered by Go, for the applause: colour fixtures in warm white at full intensity, moving heads in white at 80 % towards the centre of the stage.

**Running order**: without the **Play the songs one after the other by themselves** box, each song waits for a Go. With the box ticked, the songs follow one another by themselves; after the last song, a Go leads to **Bows**.

**The fixtures used**: DreamShow looks at each group in the venue (based on its first fixture). Moving-head groups receive the movements; groups of colour fixtures and those that only have an intensity (a PC on a dimmer, for example) receive the colours and the pulses. If the venue has no usable group, it works fixture by fixture.

Never switched on by the wizard:

- strobes: a flash is always the operator's decision;
- machines and devices marked **no direction**, **invisible** or **manual** in the venue (smoke, fans, UV, relays…), as well as lasers and smoke machines recognised by their type.

## Theatre play

### Page 3: The cues of the play

"A typical cue list is already filled in: rename, reorder, delete, add. Each cue becomes a scene triggered by Go; moving heads, if any, aim at the chosen zone and preset in the dark."

Each line is a cue. The columns:

- **Cue**: the name of the scene that will be created.
- **Mood**: the light for the cue (see the table below): **Full light**, **Day**, **Evening**, **Night**, **Warm interior**, **Cold exterior**, **Mystery**, **Audience in**, **Blackout**.
- **Zone**: the part of the stage that is lit: **Whole stage**, **Stage right**, **Centre**, **Stage left**, **Downstage**, **Upstage**.
- **Fade (s)**: the duration, in seconds, of the fade that brings in this cue (from 0 to 600; both comma and point are accepted as the decimal separator).

- **Note for the operator**: free text shown in the cue window during the performance (for example the line that triggers the Go).
- **Sound**: a sound played at the start of the cue. The **Add…** button opens the file picker (mp3, wav, m4a, aac, wma, flac); it then shows "♪" and the file name. Right-clicking the button removes the sound (tooltip: "A sound played at the start of this cue (right-click: remove)").
- ↑ and ↓: move the cue up or down one place. ✕: deletes the cue.

- **+ Cue** (below the list): adds a cue at the end, named "Scene" followed by its number, with the **Day** mood, on the **Whole stage**, with a 3-second fade.

The typical cue list offered:

| Cue | Mood | Zone | Fade (s) | Note for the operator |
|---|---|---|---|---|
| Audience in | Audience in | Whole stage | 3 | Audience coming in. Go when the house is full and the announcement made. |
| House to black | Blackout | Whole stage | 4 | Go: blackout, the actors take their places. |
| Scene 1 — day | Day | Whole stage | 3 | Go as soon as the actors are in place. |
| Scene 2 — interior | Warm interior | Centre | 3 | Go on the line where the place changes. |
| Scene 3 — evening | Evening | Whole stage | 8 | Slow: daylight fades during the scene. |
| Monologue | Mystery | Downstage | 2 | Go when the actor steps forward. |
| Scene 4 — night | Night | Whole stage | 5 | |
| Final blackout | Blackout | Whole stage | 3 | Go on the last line. |
| Bows | Full light | Whole stage | 2 | Go at the first applause. |
| Audience out | Audience in | Whole stage | 4 | |

If no cue is left, **Create the show** ✨ does nothing.

### What DreamShow writes for a play

**One scene per cue**, in order, each triggered by Go, with its cue note. The light is set in the **final state** of each scene, fixture by fixture. If a sound has been chosen, the scene gets a **Sound** track with the file at the start of the timeline; the file is copied into the show's media folder.

For each mood, DreamShow distinguishes three families of fixtures: those that light the stage, those at the back (placed in the rear half of the stage, or less than 1.5 m above the floor: floor lights, back wall, cyclorama) and the moving heads.

| Mood | Stage | Back | Moving heads |
|---|---|---|---|
| Full light | warm white, 100 % | warm white, 90 % | white, 90 % |
| Day | warm white, 85 % | sky blue, 70 % | warm white, 70 % |
| Evening | orange, 65 % | violet, 55 % | amber orange, 60 % |
| Night | night blue, 30 % | deep blue, 35 % | light blue, 45 % |
| Warm interior | warm orange, 75 % | orange, 40 % | warm white, 60 % |
| Cold exterior | bluish white, 75 % | blue, 60 % | bluish white, 60 % |
| Mystery | violet, 30 % | dark violet, 40 % | white, 90 % |
| Audience in | warm white, 45 % | off | off |
| Blackout | off | off | off |

**White-only fixtures** (a PC on a dimmer, for example) cannot turn blue: for **Night**, **Mystery** and **Evening**, DreamShow lowers their intensity instead (60 %, 50 % and 85 % of the value in the table).

**The zone**: DreamShow calculates where the beam of each colour fixture lands, just like the simulation. Fixtures whose beam lands in the zone take the intensity from the table; the others drop to 12 % of that intensity. A front light hung in the house whose orientation has not been declared is considered to cover the full width for **Stage right**, **Centre** and **Stage left**. With **Whole stage**, all fixtures take the intensity from the table.

**Moving heads** genuinely aim at the centre of the zone (using their calibration if there is one, otherwise from their position, like the aiming tool). With **Whole stage**, they spread across the width of the stage. During a **Blackout** or **Audience in** cue, they stay off but already turn towards the zone of the next lit cue: they preset in the dark.

**The fade**: in DreamShow, the fade belongs to the scene you are leaving. The fade you entered for a cue is therefore stored in the scene before it. The fade of the very first cue is not used: on the first Go of the performance, the light of the first cue comes up without a fade.

Fixtures marked **Out of order** in the venue are ignored, as are strobes and devices marked **no direction**, **invisible** or **manual**.

## The last page: Creating the show

DreamShow shows its progress ("Listening to song 2 of 3: …", then "Writing the light…"), then **Your show is ready!** followed by a summary:

- one song: the tempo (BPM), the number of sections, the duration, and a reminder that at the end Go moves on to the "Bows" scene;
- several songs: the number of songs, the total duration and, depending on your choice, "The songs follow each other by themselves." or "Go (space bar) moves on to the next song.";
- a play: the number of cues, each triggered by Go, and a reminder that each cue's note is shown in the cue window.

The **Watch and listen to the show ▶** button (musical show) or **See the show ▶** (play) closes the wizard, opens the show, switches the simulation to 3D, selects the first scene and plays it from the start.

If something goes wrong, the message "Creation failed:" followed by the reason is displayed, and the error is recorded in the **Log**.

## Where the created files are stored

- The show: `<nom du spectacle>.show.json` in the DreamShow data folder. If that name already exists, DreamShow adds "(2)", "(3)"…: an existing show is never overwritten.
- The songs and sounds: copied into `medias\<nom du show>\`; your original files are not touched.
- The ready-made venue: `<nom de la salle>.venue.json`, created only once.

See Shows, files and projects.

## Step by step

### A first show on a song

1. Menu **? > Welcome and first steps…**, tile **♫ Create my first show**.
2. Leave **A musical show** selected, click **Next →**.
3. Leave **Small stage** selected, **Next →**.
4. **Add songs…**, choose your song, **Next →**.
5. Choose **Energetic**, then **Create the show ✨**.
6. Wait for **Your show is ready!**, then click **Watch and listen to the show ▶**.
7. At the end of the song, press the space bar: the **Bows** scene lights up.

### A concert of several songs played back to back, on your own fixtures

1. First declare your fixtures: see Declaring your fixtures (venue wizard).
2. Open the wizard, choose **A musical show**, type the name of the concert in **Show name (optional)**.
3. Choose **My current venue: …**.
4. Add the songs, put them back in order with ↑ and ↓, tick **Play the songs one after the other by themselves (otherwise Go between songs)**.
5. Choose the mood, **Create the show** ✨, then **Watch and listen to the show** ▶.

### A theatre play

1. Open the wizard, choose **A theatre play**, name the play.
2. Leave **Small theatre (no moving heads)** selected or choose your venue.
3. In **The cues of the play**, rename the cues to match your cue sheet, change the moods and zones, set the fades, write the Go notes, add the sounds (**Add…**).
4. Delete the cues you do not need (✕), add more with **+ Cue**, reorder with ↑ ↓.
5. **Create the show** ✨, then **See the show** ▶.
6. To run the play, open Show mode: see Show mode and the cue window.

## Good to know

- Everything the wizard writes can then be edited like any ordinary show: click a block in the timeline to change a colour, an effect or a fade; select a scene to set its final state, its fade or its note. See The scene timeline and Scene settings.
- If you close the wizard on the last page without clicking the final button, the show is saved anyway: open it with **File > Open a show…**.
- Analysing a long song can take a few seconds; the window cannot be closed during that time.
- A ready-made venue uses specific fixture models from the library. To drive your real fixtures, declare them with the venue wizard, then run the first show wizard again on **My current venue**.

## See also

- First steps
- Declaring your fixtures (venue wizard)
- Creating a show with an AI
- Dynamic effects
- Fades and mixes
- Aiming and covering a zone with moving heads
- Show mode and the cue window

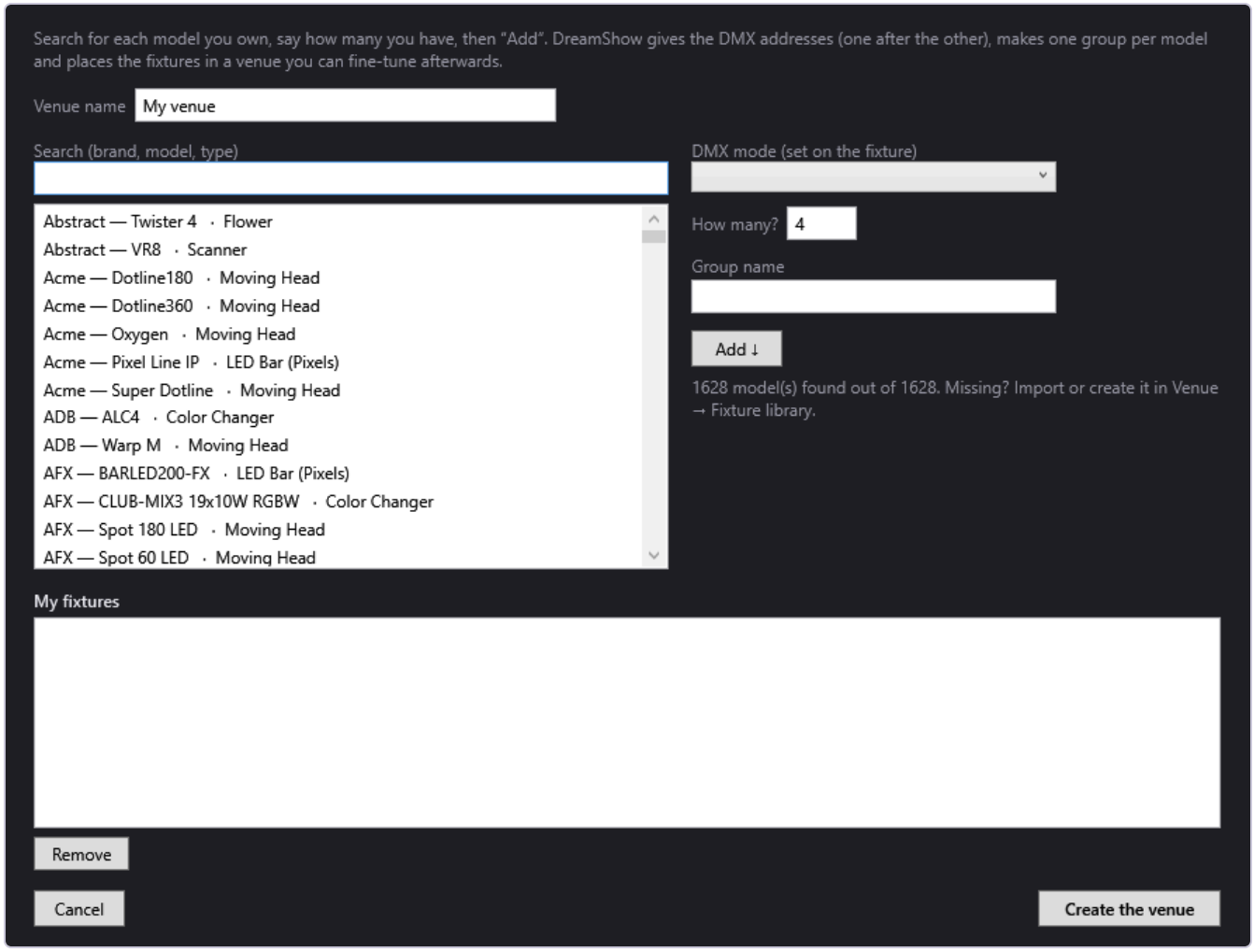
# Declaring your fixtures (venue wizard)

The **Declare my fixtures** wizard creates a venue with your real equipment in a minute: you look up each fixture model you own, and give its DMX mode and how many you have. DreamShow assigns the DMX addresses, creates one group per model and places the fixtures on a venue plan that you will fine-tune afterwards.

**To open it:**

- menu **? > Welcome and first steps...**, tile ⚙ **Declare my fixtures**;
- or the **Declare my fixtures...** button that appears in the middle of the simulation when the active venue has no fixture.

If the open show has unsaved changes, DreamShow first asks **Save the changes to the show?**.




## The window

At the top, a reminder: “Search for each model you own, say how many you have, then “Add”. DreamShow gives the DMX addresses (one after the other), makes one group per model and places the fixtures in a venue you can fine-tune afterwards.”

- **Venue name**: the name of the venue to create. By default “My venue”. If left empty, it is also “My venue”. It must be different from the venues that already exist.

### On the left: finding a model

- **Search (brand, model, type)**: type one or more words. The list only keeps the models that contain all the words, in the brand, the model or the type (for example `par rgbw`, `moving head`, `beam 230`). Upper and lower case make no difference.
- **The results list**: each line reads “Brand — Model · Type”. At most 300 results are shown: narrow down the search if your model does not appear. Double-clicking a line does the same as **Add ↓**.
- Below the **Add ↓** button, a line gives the number of models found: “12 model(s) found out of … . Missing? Import or create it in Venue → Fixture library.”

The list comes from the fixture library of the active venue. See The fixture library.

### On the right: setting up the chosen model

- **DMX mode (set on the fixture)**: the list of modes of the selected model, with their number of channels, for example “10 Channel (10 channels)”. Choose the mode set on your fixtures (in the fixture's own menu): the addresses depend on it. By default, the first mode in the list.
- **How many?**: the number of fixtures of this model, from 1 to 200. 4 by default.
- **Group name**: the name of the group that will gather these fixtures. It is filled in with the model name when you select one; you can change it (for example “Front”, “Backlights”, “Heads stage right”). If left empty, the model name is used.
- **Add ↓**: adds the line to **My fixtures**.

Possible messages:

- “First choose a model in the list.”: no model (or no mode) selected.
- “Invalid number (1 to 200).”: the **How many?** field is not a number between 1 and 200.
- After each addition: the total, for example “10 fixture(s), 136 DMX channels in total.”

### At the bottom: My fixtures

- **My fixtures**: one line per addition, for example “4 × Brand Model (10 Channel) → group « Front »”. You can add the same model several times (in another mode or another group).
- **Remove**: removes the selected line.
- **Cancel**: closes the wizard without creating anything.

- **Create the venue**: creates the venue. If the list is empty: “Add at least one fixture.” If a venue already has that name: “A venue “…” already exists: choose another name.”

## What DreamShow creates

### DMX addresses

The fixtures are addressed one after the other, in the order of the **My fixtures** list, starting from address 1 of universe 1. Each fixture takes up the number of channels of its mode. A fixture is never split across two universes: if it no longer fits within the 512 channels, it starts at address 1 of the next universe. The number of universes of the venue is set accordingly.

Example: 4 fixtures in 10 channels, then 2 moving heads in 18 channels → addresses 1, 11, 21, 31, then 41 and 59, all in universe 1.

### Groups and names

- One group per line of **My fixtures**, with the chosen name. If two lines have the same group name, the second one becomes “Name 2”.
- Each fixture is named after its group: “Front 1”, “Front 2”…

### The venue plan

DreamShow draws a venue sized to your rig:

- width: 4 m plus 0.9 m per fixture, between 8 and 24 m; length: 1.3 times the width; height: 6.5 m; stage depth: 0.4 times the width;
- each line of **My fixtures** forms an evenly spaced row across 80 % of the width, 5.5 m high;
- the first moving-head model goes on an upstage bar; the following moving-head models and the strobes on a mid-stage bar;
- all other fixtures (PARs, washes, profiles, dimmers…) go on a front-of-house bar, 1.5 m in front of the edge of the stage.

### After “Create the venue”

1. The venue is saved in the data folder (`<nom de la salle>.venue.json`) and becomes the active venue.
2. An empty show is created for it, with the venue's name and a single scene “Scene 1”, and it is opened: no show refers to the old venue any more.
3. The simulation switches to 3D.
4. A message sums up: “Your venue “…” is ready: … fixtures, grouped by model.” It reminds you to set up the interface in **Venue > DMX connections…** and the positions in **Venue > Fixtures, addresses, positions…**, then asks: “Create a first show on a song now, with these fixtures?”

- **Yes**: the **Create my first show** wizard opens, with **My current venue** already chosen. See Creating your first show (music or theatre).
- **No**: you stay on the empty show.

The creation is recorded in the **Log**, with the number of fixtures and universes.

## Step by step

### Declaring a small rig

1. Open **? > Welcome and first steps…** and click ⚙ **Declare my fixtures**.
2. In **Venue name**, type the name of your venue.
3. In **Search (brand, model, type)**, type the brand or model of your PARs.
4. Click the right model in the list.
5. In **DMX mode (set on the fixture)**, choose the mode shown on your fixtures.
6. In **How many?**, type how many there are; change **Group name** if you wish (for example "Front").
7. Click **Add** ↓.
8. Repeat steps 3 to 7 for your moving heads, then for each other model.
9. Check **My fixtures** and the channel total, then click **Create the venue**.
10. Note the addresses assigned (in **Venue > Fixtures, addresses, positions…**) and set each fixture to its address.

### Next, to drive the real fixtures

1. Connect your DMX interface (USB or network node).
2. Set it up in **Venue > DMX connections…**: see DMX connections.
3. Adjust the real positions and the orientation of the fixtures in **Venue > Fixtures, addresses, positions…**: see Fixtures, addresses, positions.

## Good to know

- The order of **My fixtures** determines the addresses: declare the models in the order of your DMX cabling if you want addresses that follow on along the line.
- If your fixtures are already addressed differently, keep their addresses: change them afterwards in **Venue > Fixtures, addresses, positions…** rather than re-addressing the whole rig.
- The plan is a starting point: heights, positions and angles must be corrected so that the simulation and moving-head aiming match reality.
- A model missing from the list can be created or imported (GDTF, QLC+, DJworld) in **Venue > Fixture library…**; then run the wizard again.
- The wizard always creates a new venue; it never modifies an existing venue.

## See also

- Fixtures, addresses, positions
- The fixture library
- DMX connections
- Switching, renaming, duplicating a venue
- Creating your first show (music or theatre)

# Installation, languages and updates

This page explains how to install DreamShow, what is installed and where, how to choose the interface language, how to find out that a newer version exists and how to install it without losing your work.

**To open it:** the **Language** menu (the globe icon), **? > Check for updates...**, **? > About DreamShow...**.

## System requirements

- Windows 10 or 11, 64-bit.
- A recent graphics card improves the 3D simulation.
- No administrator rights are needed to install DreamShow.
- No Internet connection is needed to work. The Internet is only used to check for updates, for the optional news subscription and to update the fixture library online.

## Installing DreamShow

1. Run the installer `DreamShow-Setup-<version>.exe`.
2. **Select Setup Language**: choose the installation language (French, English, German, Spanish, Italian or Chinese).
3. **Select Destination Location**: "Where should DreamShow be installed?". The folder offered is in your Windows profile (`%LOCALAPPDATA%\Programs\DreamShow`), which avoids asking for administrator rights. Click **Next**.
4. **Select Additional Tasks**: the **Create a desktop shortcut** box is unticked by default.
5. **Ready to Install**: click **Install**. If Microsoft's Visual C++ component is missing from the PC, the installer installs it too; Windows may then ask for permission.
6. **Completing the DreamShow Setup Wizard**: leave the **Launch DreamShow** box ticked to start it straight away, then click **Finish**.

### Where DreamShow is installed

The installation is for the current Windows user, in their personal programs folder (usually `C:\Users\<vous>\AppData\Local\Programs\DreamShow`). This folder is writable: DreamShow also keeps your shows and settings there, in the `data` subfolder.

| Folder or file | Contents |
| --- | --- |

| | |
|---|---|
| `DreamShow.exe` | the software (it contains everything it needs, nothing else to install) |
| `data\` | your data folder: venues, shows, media, effects, trajectories, settings (see Shows, files and projects) |
| `data\fixtures\` | the library of fixture definitions |
| `data\lang\` | the interface translation catalogues |
| `tools\` | the companion tools: the ppvideoplayer video player (with the VLC playback modules; VLC does not need to be installed) and MaskCtl, the Bluetooth driver for LED masks |
| `tools\dsplayer\` | the standalone player for Windows, Raspberry Pi and ESP32 (see The standalone player) |
| `licenses\` and `THIRD-PARTY-NOTICES.md` | the licences of third-party components |
| `DreamShow-Manuel.docx` | the manual in Word format |

The installer also adds the demonstration (in French and in English) to `data`: the demo venue and its show “DreamShow demonstration”, a large venue with 120 fixtures and its show, and the show “Showtime opening” written by an AI, with their sounds and trajectories.

In the Start menu, the **DreamShow** folder contains: DreamShow, the manual, and the uninstaller.

On a PC with two graphics cards (integrated and dedicated), the installer tells Windows to run DreamShow on the more powerful card. DreamShow also checks this setting each time it starts; if it has just written it, it restarts once by itself, before showing its window (the first start then takes a little longer).

## The first start

1. A small start-up window shows the steps: “Loading the venue and the show…”, “Preparing the interface…”, “Starting the DMX engine…”.
2. **The language**: the first time it starts, DreamShow uses the Windows display language if it is one of its six languages (French, English, German, Spanish, Italian, Chinese); otherwise, it starts in English.
3. **The show that opens**: the demonstration in the interface language (in French or in English; for the other languages, the English demonstration).
4. **The welcome screen Welcome to DreamShow** opens, then the first steps checklist. See First steps.

On later starts, DreamShow reopens the last show you worked on, with its venue.

## The Windows firewall

DreamShow uses the network for Art-Net and sACN output, Visualiser mode, the web remote, the OSC remote, the follow-spot tablets, the backstage displays and the IP video players. The first time DreamShow opens one of these functions, Windows may ask you to allow DreamShow on the network. Accept, at least for private networks, otherwise devices on the network will not be able to reach DreamShow.

If you refused by mistake, allow DreamShow in the Windows firewall settings.

## The Language menu

The language menu has no written name: it is a **globe**, so that everyone can find it even in an interface in a language they cannot read. Its tooltip reads **Langue / Language**.

It lists the six languages, each written in its own language: **English**, **Deutsch**, **Español**, **Français**, **Italiano**, **中文**. The current language is ticked.

When you choose another language, DreamShow saves the choice and asks: “Language: … The change takes effect when DreamShow restarts. Restart now?”

- **Yes**: DreamShow closes cleanly (it offers to save the show if it has been changed) and restarts in the new language.
- **No**: you carry on in the current language; the new language will apply the next time it starts.

DreamShow never changes language on the fly: there is never a forced restart, so that the show control software is never closed in the middle of a performance.

## Checking for updates

**? > Check for updates…** compares your version with the latest published version. Nothing is ever downloaded or installed without your consent.

- If you already have the latest version: “You have the latest version (…).”
- If a newer version exists: a message says “DreamShow … is available (you have …).”, followed by what has changed in that version, then asks **Open the download page?**. **Yes** opens the page in your browser.

### The automatic check

At startup, DreamShow checks by itself, at most about once a day (not until 20 hours after the previous check), whether a newer version exists. It only reads the number of the latest version. If a more recent version exists:

- the menu item becomes **New version … available…**;
- a line is added to the **Log**;
- a brief message appears for a few seconds next to the show title: “DreamShow … is available: menu ?”.

Without the Internet (an offline show control PC), the check fails silently.

### Anonymous usage statistics

DreamShow is free; to know how many people use it, it sends a small anonymous message at start-up, then at most once a day: a number drawn at random on this PC (renewed every year), the version, the language, the Windows version, and counters (time of use, number of GOs, of AI assistant requests and of actions by a connected AI, number of DMX universes set up). No show, venue or file name, nothing that identifies you, and no advertising.

To stop it: menu **?** → untick **Anonymous usage statistics**. Nothing is sent any more.

## Installing an update

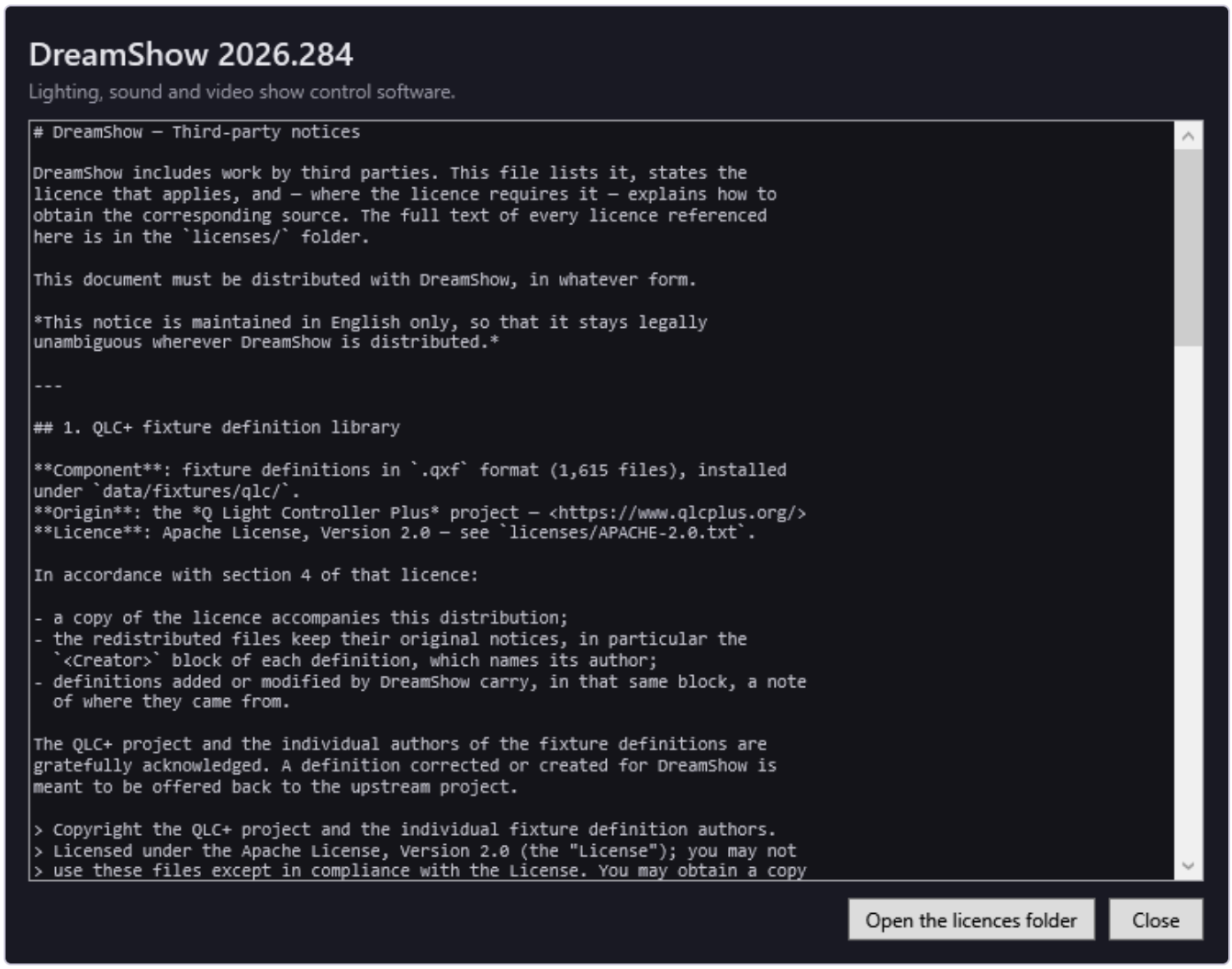
1. Save your show (**File > Save**) and, to be on the safe side, export your project: **File > Export the whole project (zip)…** (see Shows, files and projects).
2. Close DreamShow.
3. Download the new version from the download page and run the installer.
4. Install over the old version, in the same place.
5. Restart DreamShow and check the version number in the title bar.

Your shows, venues, media, palettes, trajectories and settings are kept.

## About DreamShow

**? > About DreamShow…** opens the **About DreamShow** window:

- at the top, “DreamShow” and the version number, then “Lighting, sound and video show control software.”;
- in the centre, the notices for third-party components and their licences (file `THIRD-PARTY-NOTICES.md`); if this file cannot be found: “THIRD-PARTY-NOTICES.md not found beside the application.”;
- **Open the licences folder**: opens the `licenses` folder in File Explorer;
- **Close**.




## Step by step

### Switching the interface to English

1. Click the globe in the menu bar.
2. Click **English**.
3. Answer **Yes** to the restart question.

### Checking you have the latest version before a performance

1. Connect the PC to the Internet.
2. Menu **? > Check for updates...**.
3. If a version is offered, read what has changed before deciding. Do not update just before a performance.

## Good to know

- **An update does not touch your work.** The demonstration files (demo shows, venues, trajectories and effects) are only installed if they do not exist yet: if you have modified them, your changes are kept. Your own shows, venues, media and effects are never replaced.
- Fixture definitions you have modified (folder `data\fixtures\custom`) and those updated online (`data\fixtures\online`) are not touched by an update.
- DreamShow is free: a licence installed by an earlier version is no longer needed. See DreamShow is free.
- The automatic update check can be turned off by putting `"checkUpdates": false` in the file `data\dreamshow.settings.json` (with DreamShow closed). The **? > Check for updates…** menu remains available.
- If the manual check cannot reach the Internet, DreamShow also displays “You have the latest version (…).”: check the connection before relying on it.

## See also

- DreamShow is free
- First steps
- Shows, files and projects
- Troubleshooting

# DreamShow is free

DreamShow is free: all its features are open, with no limit and no time limit. There is no licence to install and nothing to buy.

## What is included

The whole software: venues, fixture library, scenes and timelines, effects, palettes, trajectories, 2D and 3D simulation, sound, creating a show with an AI, the cue window (show mode), remotes, MIDI, OSC, timecode, sound-to-light.

And control of the real equipment:

- DMX output on all the venue's universes (Art-Net, sACN, USB DMX PRO, uDMX): see DMX connections;
- IP video players: see Video screens;
- export for a standalone player: see The standalone player;
- follow spot tablets: see Follow spot on a tablet;
- backstage displays and the teleprompter: see Backstage displays;
- humanoid robots, LED masks and connected sockets: see Humanoid robots, LED masks and Connected sockets;
- the AI assistant, which uses your own Claude API key, entered in its window: see AI assistant (set the scene);
- the MCP connector: see Connect an AI (MCP).

## A licence installed by an earlier version

If an earlier version of DreamShow had you install a licence (file `licence.txt` in the `DreamShow` folder of your Windows account's local data, `%LOCALAPPDATA%\DreamShow`), it is no longer needed: you have nothing to do.

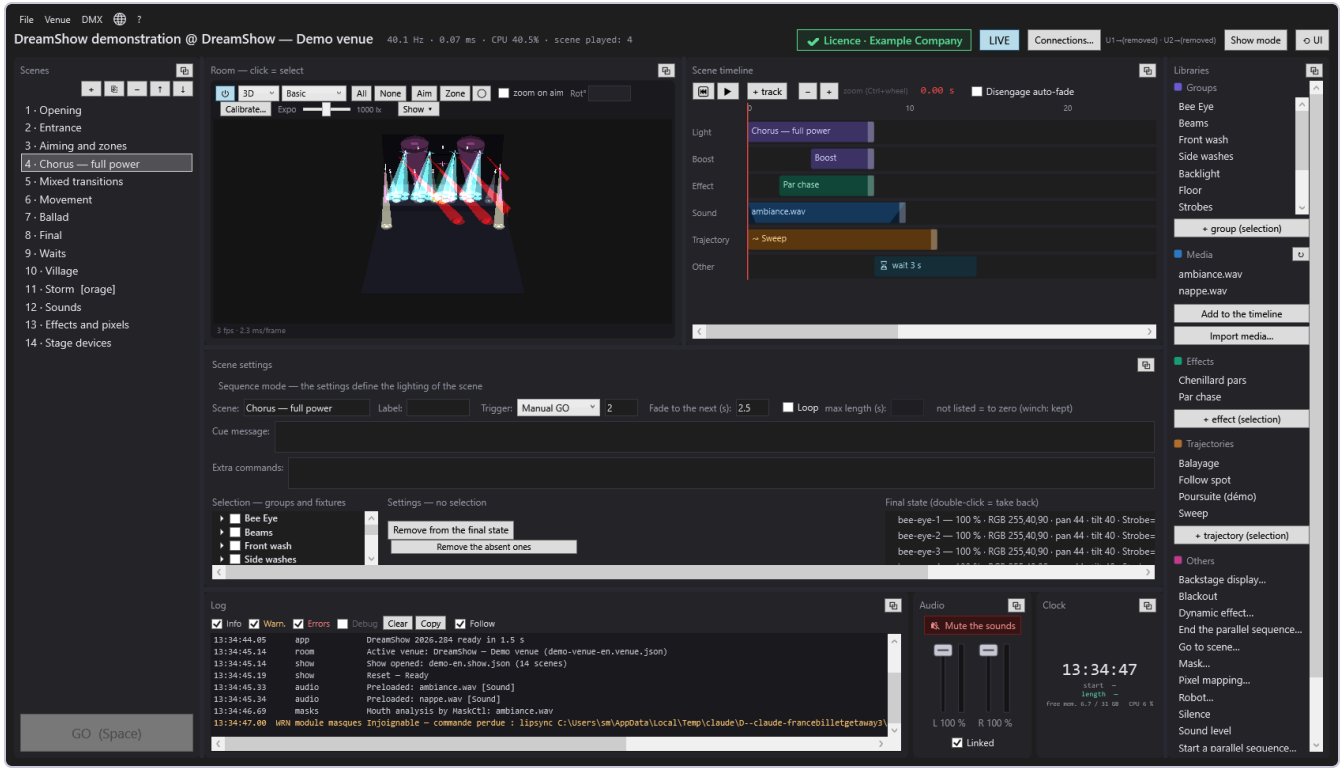
## See also

- Installation, languages and updates
- DMX connections
- The standalone player
- Follow spot on a tablet
- Backstage displays
- Troubleshooting

# Main screen overview

DreamShow's main screen is the editor: this is where you write the show, scene by scene, adjust the light, place sound and video on the timeline and see the result in the simulation. This page presents each area of the screen and points to the detailed pages.

**To open it:** it is the window that opens when DreamShow starts.




## The Windows title bar

At the very top, the window's title bar always shows:

- the DreamShow version (for example “DreamShow 2026.282”);
- the file name of the open show (for example `Mon spectacle.show.json`), or “new show”;
- the name of the active venue;
- possibly some notes in square brackets: “[visualiser — output off]” when Visualiser mode is active, “[sound-to-light]” when sound-to-light is running, “[timecode]” when the show follows a timecode.

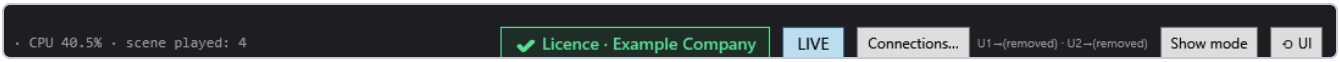
## The menu bar

Below the title bar, five menus:

- **File**: create, open and save shows (**New show…**, **Open a show…**, **Recent shows**, **Save** (Ctrl+S), **Save as…**), **Remote (GO keys)…**, **Open the show's media folder**, **Export the whole project (zip)…**, **Import a project (zip)…**, **Create a show with an AI…**, **DMX stream by an AI…**, **Export for a standalone player…**, **Import a ppshow show…**. See Shows, files and projects, The remote (GO keys), Creating a show with an AI, DMX stream by an AI and The standalone player.
- **Venue**: everything that describes the place. **DMX connections…** (DMX connections), **Manual controls…** (Manual controls), **Follow spot (tablets)…** (Follow spot on a tablet), **Fixtures, addresses, positions…** (Fixtures, addresses, positions), **Fixture library…** (The fixture library), **Screens…** (Video screens), **Robots…** (Humanoid robots), **LED masks…** (LED masks), **Backstage displays…** (Backstage displays), **Import the calibrations (lyrescal.ini)…** and **Export the calibrations…** (Moving-head calibration), then **Change venue**, **Rename the venue…**, **Duplicate the venue…** and **Delete the venue…** (Switching, renaming, duplicating a venue).
- **DMX**: **DMX values…** and **DMX faders…** (DMX values and DMX faders), **Visualiser mode (Art-Net / sACN in)…** (Visualiser mode), **Sound-to-light and tempo…** (Sound-to-light and tempo), **Timecode (LTC / MTC)…** (Timecode), **MIDI controllers…** (MIDI controllers), **Web remote…** (The web remote), **OSC remote…** (The OSC remote).
- **Language**: this menu shows no word but a small globe icon (tooltip “Langue / Language”), so that you can find it even when facing an interface in a language you cannot read. It lists the languages, each written in its own language, with the current language ticked. See Installation, languages and updates.
- **?**: **Welcome and first steps…** and **First steps checklist…** (First steps), **Check for updates…** (Installation, languages and updates), **About DreamShow…** (version and third-party component notices).

## The top bar

Below the menus, a bar brings together the show status on the left and the general commands on the right.




### On the left: the show and the statistics

- **Show name @ venue**: in large type, the name of the show followed by “@” and the name of the venue (for example “Concert du 14 @ Petite scène”).
- **Asterisk**: an orange “*” appears after the name as soon as the show contains unsaved changes. It disappears when you save.

- **Save confirmation**: after saving, the green text "✓ show saved (hh:mm:ss)" is displayed for about two and a half seconds.
- **Statistics** (status line): a line in small type, updated twice a second:
  - the actual rate of the DMX engine in Hz (the number of DMX frames computed per second);
  - the average time to compute one frame, in milliseconds;
  - "CPU … %": the share of the processor used by DreamShow;
  - "scene played: …": the number of the scene the engine is playing ("—" when no scene is running, for example in the "Ready" state before the first Go);
  - "· BLACKOUT" when the general blackout is active;
  - "· ⇉ in parallel: …" with the list of the parallel sequences running.

This same line also displays, very briefly, some messages (for example "Aim: select one or more moving heads first", "Calibrate: select a moving head", "3 media imported into medias\…") or a show compilation error ("compilation error: …").

### On the right: the general commands

From left to right:

- **Simple mode**: a two-state button, for beginners. When on, the screen shrinks to the essentials: the scenes, the venue, the timeline and the scene settings. The **DMX** menu and the advanced functions (remote control, prompter, project export, robots, screens, masks, sockets, displays, calibrations, venue management) are hidden, as are the venue's **Aim**, **Zone**, **Calibrate…** and **Expo** tools, the **Libraries** and the bottom strip (log, audio, clock). Switch it off and everything comes back. Nothing is deleted or converted: the show is the same in both modes. Simple mode is offered on a first installation; an existing installation stays in full mode. Your choice is remembered.
- **LIVE**: a two-state button, on at startup.
  - On: the scene selected in the **Scenes** list is sent live to the DMX interfaces (Art-Net, sACN, USB, uDMX). Selecting another scene makes the engine jump to it, and every adjustment you make goes straight out to the fixtures.
  - Off: no more frames go out to the interfaces. The simulation carries on. Selecting a scene no longer moves the engine: the simulation then shows the final state of the selected scene, computed separately.
  - In Show mode (cue window open), **LIVE** has no effect: the output is always active, so that an unlucky click cannot plunge a performance into darkness.
  - In Visualiser mode, DreamShow sends nothing, whatever the state of **LIVE**.
- 🤖 **disarmed** / 🤖 **ARMED**: only appears if the venue declares at least one robot. Disarmed, the Robot blocks of the timeline are ignored (you rehearse light and sound without the robot moving); armed (red background), they are executed. The state of the first robot follows the label ("· hors ligne" (offline), "· not found"…). See Humanoid robots.

- **Connections…**: opens the same window as **Venue > DMX connections…** (universes and interfaces of the current venue). When you confirm, the connections are saved in the venue file and the engine restarts on the new outputs. See DMX connections.
- **Outputs in service**: just to the right of the button, a small grey text sums up the open outputs, universe by universe, for example “U1→Art-Net to 192.168.1.50”; several universes are separated by “·”. “U3→(removed)” flags a universe used by the venue but with no output. The tooltip repeats the full text and the errors from opening the interfaces.
- **Show mode**: opens the cue window, which you can place on a second screen. See Show mode and the cue window.
- ⟲ **UI**: restores the default panel layout (see below).

## The panels

The rest of the window is divided into eight panels. Each one has a title in small grey type at the top left, and a ⧉ button at the top right.

| Panel (exact title) | Default position | Role | Detailed page |
|---|---|---|---|
| **Scenes** | left column | the list of the show's scenes, the buttons to add, duplicate, delete and move them, and the big **GO** button | The Scenes panel |
| **Room — click = select** | top centre, left | the 2D or 3D simulation of the venue, fixture selection, the Aim and Zone tools | 2D and 3D simulation |
| **Scene timeline** | top centre, right | the tracks and blocks of the selected scene, the red playback line | The scene timeline |
| **Scene settings** | middle | the scene's name, label, trigger, fade, loop, message and commands, then fixture selection, the sliders, the colour palette, the specific settings and the final state | Scene settings and Selecting fixtures and adjusting the light |
| **Libraries** | right column | the groups, media, effects, trajectories and other events to drag onto the timeline | Libraries |
| **Log** | bottom, left | DreamShow's messages | Log, Audio and Clock |
| **Audio** | bottom, middle | mute all sounds, left and right volumes, level meters | Log, Audio and Clock |

| | | | |
|---|---|---|---|
| **Clock** | bottom, right | the time, the start and duration of the show, memory and processor | Log, Audio and Clock |

## Arranging the panels

### Resizing

Between the panels, thin grey strips act as splitters: drag them to enlarge a panel at the expense of its neighbour. Each panel has a minimum size.

### The simulation / timeline split

The simulation (**Room — click = select**) and the **Scene timeline** share the top row, half each by default. The vertical splitter between the two sets the proportion, which is remembered. When one of the two panels is detached into a floating window, the other takes the full width of the row.

### Swapping two panels

Click a panel's title (the cursor turns into a four-arrow cross), hold the button down and release over another panel: the two panels swap places. For example, you can put the **Scene timeline** in the large middle area and the **Scene settings** at the top.

### Detaching a panel into a floating window

- The ⧉ button (tooltip “Detach into a floating window”) takes the panel out into a separate window, titled “DreamShow —” followed by the panel title. The space it took up is given back to the other panels.
- You can move this window to another screen and resize it. It stays on top of the main screen.
- To put the panel back in its place, simply close its floating window.

### Remembered layout

The layout is saved automatically in the `dreamshow.ui.json` file in the data folder: window size (and whether it was maximised), position of each panel, splitter sizes, floating windows (position and size). The next time DreamShow starts, everything is put back as you left it, floating windows included. The same file also keeps some simulation settings: categories displayed, background plan, exposure, 3D quality and the **zoom on aim** box.

### Going back to the default layout: ↺ UI

The ↺ **UI** button (tooltip “Restore the default layout”):

1. closes all the floating windows and puts their panels back in the main window;

2. puts each panel back in its original position (see the table above) and at its original size;
3. deletes the layout file.

It does not change the simulation settings (categories displayed, exposure, 3D quality, zoom on aim): they are saved again the next time the layout is saved, for example when DreamShow closes.

## The editor's GO button

At the bottom of the **Scenes** panel, the big **GO (Space)** button moves on to the next scene, just as during the performance. The space bar does the same, except when the cursor is in a text box, a check box or a drop-down list.

- The button is greyed out as long as a Go is not possible: on the last scene, or as long as the timeline of the current scene has not played to the end (the red line must have reached the end of the last block, and no fade must be in progress). It is always active on a scene set to **Loop** (Go leaves the loop) and before the first scene (“Ready” state).
- Go applies the fade of the scene you are leaving (**Fade to the next (s)**), then plays the timeline of the scene you arrive at.
- If the timeline's **Disengage auto-fade** box is ticked, Go arrives on the next scene **without** playing its timeline: the scene is shown frozen on its final state, with no sound and no blocks. This lets you step through the show by hand.

The details are in The Scenes panel.

## Main screen shortcuts

| Key | Effect |
|---|---|
| Space | Go (same rule as the **GO** button) |
| Ctrl+S | save the show |
| Ctrl+wheel (on the timeline) | timeline zoom |
| Delete (in the **Trajectories** list) | delete the selected trajectory |

When a “Wait for a key…” block is waiting for a key in the editor, that key takes priority to release the wait. See Waits and Keyboard shortcuts.

## Step by step

### Making the timeline large and putting the simulation on a second screen

1. Click ⧉ at the top right of the **Room — click = select** panel.
2. The simulation opens in its own window; the timeline now takes up the whole top row.

3. Drag the simulation window onto the second screen and maximise it.
4. The next time DreamShow starts, the simulation window will reopen in the same place.
5. To put it back in the main screen, close its window.

### Swapping the timeline and the scene settings

1. Click the **Scene timeline** title and hold the button down.
2. Drag it onto the **Scene settings** panel and release.
3. The two panels have swapped places.

### Getting back to a clean layout

1. Click ↻ **UI** at the top right.
2. All the panels go back to their original position and size.

## Good to know

- Closing DreamShow always asks for confirmation (“Quit DreamShow?”). If the cue window is open, the question becomes “A show is RUNNING. Quit DreamShow anyway?”. If the show has unsaved changes, DreamShow then offers to save them.
- Panels keep working when they are detached: the simulation, the timeline and the log follow the show as they do in the main window.
- If the simulation slows the PC down, turn off its rendering with the ⏻ button of the Venue panel: the DMX output is not affected. See 2D and 3D simulation.
- The engine keeps playing the show even when a settings window is open.

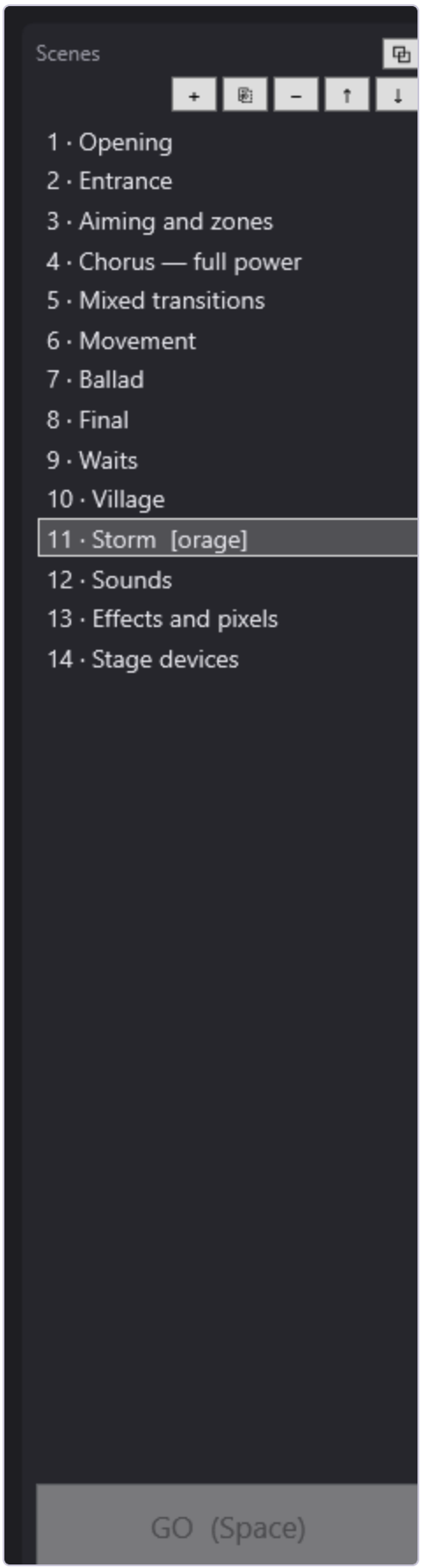
## See also

- The Scenes panel
- Scene settings
- Selecting fixtures and adjusting the light
- 2D and 3D simulation
- The scene timeline
- Libraries
- Log, Audio and Clock
- Show mode and the cue window

# The Scenes panel

A DreamShow show is a sequence of scenes, played one after the other on Go, automatically or chained. The **Scenes** panel lists them: this is where you add, copy, delete and arrange scenes, choose the one to edit, and test the move from one scene to the next with the **GO** button.

**To open it:** the panel is in the left column of the main screen (default layout).




## The scene list

Each line shows:

- the number of the scene in the show (1, 2, 3…);
- a dot "·" then the name of the scene;
- if the scene has a label, that label in square brackets, for example `4 · Orage [pluie]`. The label is a short, stable name used by jumps and parallel sequences (see Scene settings and Jumps and scene labels).

The name and the label are edited in the **Scene settings** panel; the list updates with every keystroke.

## The buttons at the top

| Button | Tooltip | What it does |
|---|---|---|
| + | Add a scene | inserts an empty scene just after the selected scene (or at the end if none is selected), named "Scene N" where N is the number of scenes plus one, and selects it |
| ⎘ | Duplicate the scene (inserted just after, named #2, #3…) | full copy of the selected scene (settings, final state, tracks and blocks), inserted just after it |
| − | Delete | deletes the selected scene, after confirmation |
| ↑ | — | moves the selected scene up one place |
| ↓ | — | moves the selected scene down one place |

### Details of each button

- **+**: the new scene has no tracks and no light settings. Its trigger is **Manual GO**, with no fade.
- ⎘: the copy has the name of the original followed by "#2". If copies already exist ("Final #2", "Final #3"), the new one takes the next number ("Final #4"). Duplicating "Final #3" also gives "Final #4": the number carries on from the highest copy of the same name.
- **−**: DreamShow asks "Delete scene "…"?". If "Go to scene…" jumps pointed to this scene, a message warns that they need fixing: they are shown in red with ⚠ on the timeline. The following scene takes the selection (the previous one if you deleted the last one).
- ↑ and ↓: swap the selected scene with its neighbour. Nothing happens on the first scene (↑) or the last one (↓).

When you insert or delete a scene with **+**, ⎘ or **−**, DreamShow renumbers the jumps ("Go to scene…") and parallel sequences that target a scene by its number, so that they keep targeting the right scene.

## Selecting a scene

A click on a scene (or the keyboard arrows in the list) selects it. The whole screen then switches to it:

- the **Scene settings** panel shows its settings (name, label, trigger, fade, loop, message, extra commands);
- the **Final state** list shows its light settings, and the sliders move to the selected target if it already has a setting;
- the **Scene timeline** shows its tracks and blocks, and goes back to the start;
- the simulation shows this scene.

If **LIVE** is on (top right), the engine jumps to this scene at once, with no fade, and the DMX output shows it live. The timeline stays paused (unless playback ▶ is running): the venue shows the final state of the scene and the blocks under the red line.

If **LIVE** is off, the engine does not move: the simulation shows the final state of the selected scene, computed separately, and nothing goes out to the fixtures.

## The GO button

At the bottom of the panel, the big **GO (Space)** button moves on to the next scene, exactly as during a performance. The space bar does the same, except when the cursor is in a text box, a check box or a drop-down list.

### When GO is possible

The button is greyed out (and the space bar has no effect) when:

- the engine is already playing the last scene;
- the timeline of the current scene has not finished: the red line must have reached the end of the last block, and no fade must be in progress.

It is always active:

- before the first scene, in the “Ready” state (Go starts scene 1);
- on a scene set to **Loop** (Go leaves the loop).

A scene with no blocks at all is “finished” straight away: Go is therefore possible immediately.

### What GO does in the editor

1. The engine moves to the scene that follows **the one it is playing** (not necessarily the one you have selected, if **LIVE** is off).
2. The fade of the scene being left (**Fade to the next (s)**) is applied: the light glides towards the final state of the new scene.
3. When the fade ends, the timeline of the new scene starts: sounds, videos, blocks, extra commands.

4. If **LIVE** is on, the **Scenes** list follows: the new scene is selected.

If the **Disengage auto-fade** box (timeline toolbar) is ticked, Go arrives on the next scene **without playing its timeline**: the scene stays paused on its final state, with no sound and no blocks, and no scene follows on by itself. This is useful for stepping through a show scene by scene while you write it.

## What happens during playback

Playback of a scene in the editor is started with the timeline's ▶ button (or by a Go). During playback:

- the red line moves along the timeline; sounds, videos and blocks are triggered when it reaches them;
- the simulation follows the red line: it reads the real DMX output, fades included;
- a **Manual GO** scene, without a loop, stops by itself at the end of its last block: the red line parks there and playback pauses, like a media player at the end of a track;
- an **Auto** or **Chained** scene moves on by itself to the next one at the planned moment (unless **Disengage auto-fade** is ticked or it is the last scene); the **Scenes** list follows if **LIVE** is on;
- a **Loop** scene restarts its timeline at the end, until Go or until its maximum duration.

In Show mode (cue window open), the cue list is in charge: the **Scenes** list follows the scene being played, and arriving on a scene always plays it. See Show mode and the cue window.

## Step by step

### Creating three scenes and testing them

1. Click **+**: “Scene 2” appears below the selected scene. Click **+** again for “Scene 3”.
2. Select each scene and set its light (see Selecting fixtures and adjusting the light).
3. Select scene 1, then in **Scene settings** set **Fade to the next (s)** to 3.
4. Press the space bar: the light glides to scene 2 in 3 seconds, and the list selects scene 2.

### Making a variant of a scene

1. Select the scene to copy.
2. Click ⎘: the copy, named for example “Salut #2”, is inserted just after it.
3. Change the copy as you wish; the original is not touched.
4. If the original had a label, give the copy a different label (see Good to know).

### Reordering scenes

1. Select the scene to move.

2. Click ↑ or ↓ as many times as needed.

## Good to know

- ⎘ also copies the scene's **Label**. With two scenes sharing the same label, jumps and parallel sequences target the first one, and the log reports it (“Label “…” used by several scenes: the first one (…) is kept.”). Change the label of the copy.
- ↑ and ↓ do not renumber jumps: a “Go to scene…” jump saved by number (scene with no label) keeps targeting the same **position** in the list, and therefore a different scene after a move. Give a **Label** to scenes targeted by jumps: the jump then follows them wherever they are.
- A jump from the editor (click in the list) is instantaneous: no fade is applied. The fade only plays on Go or during an automatic follow-on.
- A greyed-out GO is normal as long as the timeline has not played to the end: start playback with ▶, or go back to a scene with no blocks.
- Any change marks the show as modified (“*” after its name). Save with Ctrl+S or **File > Save**.

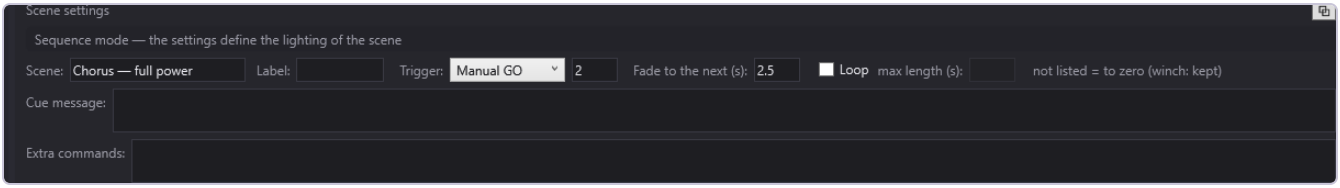
## See also

- Scene settings
- The scene timeline
- Jumps and scene labels
- Fades and mixes
- Show mode and the cue window
- Main screen overview

# Scene settings

The top of the **Scene settings** panel describes the selected scene: its name, its label, how you leave it (on Go, automatically or chained), the fade to the next scene, the loop, the message shown to the operator and the commands to run when it starts. The bottom of the same panel (fixture selection, sliders, final state) is described in Selecting fixtures and adjusting the light.

**To open it:** select a scene in the **Scenes** panel; its settings appear in the **Scene settings** panel (in the middle of the main screen in the default layout).




## The mode banner

At the very top of the panel, a banner reminds you what the light settings change:

- “Sequence mode — the settings define the lighting of the scene” (grey background): this is the normal mode. The sliders and the palette write into the final state of the selected scene.
- “EFFECT MODE — “name”: the settings change the effect, not the sequence” (orange background): you are editing an effect from the library (double-click an effect in **Libraries**). See Libraries.
- “TRAJECTORY MODE — “name”: the settings change the selected key frame” (orange background): you are editing a keyframe of a trajectory. See Trajectories.

In both orange modes, a **✓ Finish editing the effect** button appears on the right of the banner; it takes you back to sequence mode.

## The settings line

All these fields are saved in the show as soon as you change them (the show is marked “*” until the next save). They can only be edited if a scene is selected.

### Scene:

- **Scene:** the name of the scene, free text. It appears in the **Scenes** list, in the cue list and in the log. The list updates with every keystroke.

### Label:

- **Label:** a short, stable name for the scene, for example `pluie` or `final` (tooltip: “Short, stable name of the scene (e.g. rain): jumps and parallel sequences target it by this name, even when scenes are inserted or moved”).
    - It is shown in square brackets in the **Scenes** list: `4 · Orage [pluie]`.
    - Spaces at the start and end are removed, as is an “@” at the start (in blocks, a label is written `@pluie`).
    - Empty = no label.
    - Two scenes must not have the same label: otherwise the first one is targeted, and the log reports it.
    - See Jumps and scene labels and Parallel sequences.

### Trigger:

The **Trigger:** list says how you **leave** this scene for the next one. Three choices:

- **Manual GO** (default value): the scene waits for the operator to press Go. Nothing happens by itself.
- **Auto**: the scene moves on to the next one by itself after the delay given in the small box just to the right (tooltip “Auto delay (s)”, 2 by default). The delay is counted in seconds **from the start of the scene's timeline**, that is once the incoming fade is over (a scene's timeline starts at the end of the fade leading to it). Both comma and full stop are accepted as the decimal separator. An unreadable value counts as 2 seconds.
- **Chained**: the scene moves on to the next one by itself as soon as its timeline has finished, that is at the end of the block that ends last. A chained scene with no blocks at all moves on to the next one immediately (as soon as its incoming fade has finished).

The small delay box stays visible whatever the choice, but it is only used with **Auto**.

In every case:

- when leaving the scene, it is **its** fade (**Fade to the next (s)**) that applies;
- **Auto** and **Chained** have no effect on the last scene of the show;
- **Auto** and **Chained** are suspended when the **Disengage auto-fade** box is ticked (timeline toolbar, or cue window): you then only move on with Go;
- a Go is still possible on an **Auto** or **Chained** scene, as soon as its timeline has finished (see The Scenes panel).

### Fade to the next (s):

- **Fade to the next (s):** the duration, in seconds, of the fade applied when **leaving** this scene (on Go or when chaining automatically), towards the final state of the next scene. Tooltip: “As in

ppshow: length of the fade applied when LEAVING this scene (on GO or automatically), towards the state of the next scene. The timeline of the next scene starts when the fade ends."
    - 0 (default) = instant change.
    - Comma or full stop accepted. An unreadable value is ignored (the old value is kept).
    - The timeline of the next scene (sounds, videos, blocks, extra commands) only starts **at the end** of the fade.
    - The first Go from the "Ready" state arrives on scene 1 with no fade. A click in the **Scenes** list in the editor is instantaneous too.
    - How each type of channel goes through the fade (intensity, colour, position, shutter…) is detailed in Fades and mixes.

### Loop

- **Loop** (check box): the scene's timeline automatically starts again from the beginning when it reaches its end (the end of the block that ends last), until Go, or until the maximum length. Tooltip: "The timeline loops automatically until GO (or until the maximum length)".
    - On each pass, all the timeline's events play again: sounds and videos restart from the beginning, the extra commands are run again.
    - A looping scene **ignores its Auto or Chained trigger**: you leave it with a Go, or through the maximum length.
    - Go is always possible on a looping scene, even in the middle of a pass.
    - A looping scene with no blocks at all does not loop (there is nothing to replay): its **Trigger** applies normally.

### max length (s):

- **max length (s):** can only be edited if **Loop** is ticked. Tooltip: "Empty = loops until GO; otherwise the scene chains to the next one after this total time".
    - Empty (or 0): the scene loops until Go.
    - A duration: after this total time (all passes counted, from the start of the timeline), the scene moves on to the next one like an **Auto** scene, with its fade. This move is also suspended by **Disengage auto-fade**.

### The "not listed = to zero" reminder

On the right of the line, the grey reminder "not listed = to zero (winch: kept)" sums up an important rule of the final state: a fixture that has no setting in a scene's final state is set to zero (switched off) there. Only fixtures set to "hold" in the venue (the "If not mentioned in a scene" field in Fixtures, addresses, positions, typically a winch) keep the value of the previous scene.

## Cue message:

- **Cue message:** free text, on several lines, shown large in the cue window during this scene. Tooltip: “Shown large during this scene in show mode — what to expect before pressing GO”. For example: “GO when the actress sits down”. See Show mode and the cue window.

## Extra commands:

- **Extra commands:** a text area (fixed-width font) for commands run when the scene starts. The syntax is that of the ppshow software.

### When they run

- They go out at the **start of the scene's timeline**, so once the fade leading to it has finished.
- Only during playback: a scene that is merely selected in the editor, with the timeline paused, does not run them. They go out on Go, when chaining, or when you start playback (▶) with the red line at the beginning.
- A **Loop** scene runs them again on every pass.

### Syntax

- One command per line.
- A line that starts with `;` or `#` is a comment, and is ignored.
- Empty lines are ignored.
- The keyword, in upper or lower case, is followed by `=` and its value.

| Command | Effect |
|---|---|
| `EXEC=<ligne de commande Windows>` | starts a program or opens a file, as Windows would. The first word is the program (put it in quotes if it contains spaces), the rest its arguments. The log writes “EXEC: …”, or an error if the launch fails. |
| `VP=<commande ppvideoplayer>` | sends the line as is to the ppvideoplayer video player, for example `VP=play:intro.mp4` or `VP=stop:intro.mp4`. |
| `HOLOFAN=<commande ppholofan>` | sends the line as is to the ppholofan program. |
| `CHAIN=<spectacle>, <n° de scène>` | chains on to another show: DreamShow opens the show file (looked for in the current show's folder; the `.show.json` extension is added if needed) and goes to the given scene (number starting from 1; 1 if missing). If the current show has unsaved changes, DreamShow first offers to save them. A show that cannot be found is reported in the log (“CHAIN: show not found …”). |
| `PPIP={[ip;ip…]} <commande>` | sends a command to the IP video boxes (ppip), which connect to DreamShow on TCP port 5555. In square brackets, the list of targeted |

| | |
|---|---|
| | boxes by the **last number** of their IP address, separated by `;` (for example `PPIP=[101;102]…`); without brackets, the command goes to all connected boxes. |
| `AMBILIGHT=<adr>{W}{@<master>}{;…}/<adr>… ou OFF` | makes fixtures follow the colours of the videos marked (AMBI): see below. |
| `REPLAYDMX=<fichier bin>` | replays a raw DMX recording. The path is relative to the project folder (the data folder that contains the venues and shows). |

The keywords of ppshow's legacy hardware (`SMOKE=`, `MISC1=` to `MISC4=`, `LOAD=`, `MATRIX=`, `RANDOM=`…), and more generally any unrecognised line, do nothing: the log shows “Unsupported ppshow command: …” at the moment the line should have run.

### AMBILIGHT in detail

A video is marked by adding `(AMBI)` to its file name, for example `ouverture (AMBI).mp4` (see Sound and video on the timeline). When the video player plays such a video, it sends DreamShow the dominant colour of the left and right of the picture. `AMBILIGHT=` says which fixtures take these colours.

- Before the `/`: the fixtures that take the **left** colour; after the `/`: those that take the **right** colour.
- Each fixture is given by the DMX address of its red channel, in the **first universe**; green and blue follow at the next two addresses.
- `W` just after the address: the fixture also has a white channel (at address +3), set to the minimum of R, G, B.
- `@<master>`: the address of the fixture's master intensity channel. Its level, set by the scene, scales the colours, and the channel itself is pushed to full.
- Several fixtures on the same side are separated by `;`.
- `AMBILIGHT=OFF` turns the tracking off.
- The setting is **kept from scene to scene** until a new `AMBILIGHT=` line.
- When no fresh colour arrives (more than half a second without a marked video), the fixtures go back to what the scene gives them.

Example: `AMBILIGHT=1W@5;20/40;60` — on the left, an RGB+W fixture at address 1 with its master intensity at 5, and an RGB fixture at address 20; on the right, two RGB fixtures at addresses 40 and 60.

### REPLAYDMX in detail

- The file is a raw DMX recording in ppshow's format, read at one frame every 10 ms, on the **first universe**.

- During playback, it **replaces** everything the scene computes on that universe, until the end of the file. The log announces the start (“REPLAYDMX: playing …”) and the end.
- A pause freezes playback; a reset of the show stops it.

## Step by step

### A pre-show scene that loops until Go

1. Select the pre-show scene in **Scenes**.
2. Place some music and a few light blocks on its timeline (see The scene timeline).
3. Tick **Loop**; leave **max length (s):** empty.
4. In **Cue message:**, write “GO when the house is full”.
5. Set **Fade to the next (s):** to 5.
6. During the performance, the music and the blocks loop; Go starts a 5-second fade to the next scene.

### Chaining three scenes on their own

1. Scene 1: **Trigger: Auto**, delay 10: it moves on to scene 2 ten seconds after it starts.
2. Scene 2: **Trigger: Chained**: it moves on to scene 3 at the end of its last block.
3. Scene 3: **Manual GO**: it waits for the operator.
4. To test without the scenes chaining, tick **Disengage auto-fade** in the timeline toolbar.

### Starting a program when a scene starts

1. In **Extra commands:**, type for example: ``` ; ouvre le diaporama de l'entracte EXEC="C:\Programmes\Diaporama\diapo.exe" entracte.txt ```
2. Start the scene (Go or ▶ at the start of the timeline): the program starts, and the log shows “EXEC: …”.

## Good to know

- The trigger says how you **leave** a scene, and the fade also belongs to the scene you are **leaving** (as in ppshow). To adjust the arrival on scene 5, set the fade of scene 4.
- To keep a scene on a fixed state for a long time without playing anything, leave it on **Manual GO** with no blocks: it holds its final state until Go.
- Waits for a time, a date, a key or sunset are not set here but with timeline blocks: see Waits.
- The `CHAIN=`, `AMBILIGHT=` and `REPLAYDMX=` commands only run in the main cue list, not in a parallel sequence.
- `EXEC=` starts any program on the PC: check the line before the performance, and look at the log if in doubt.

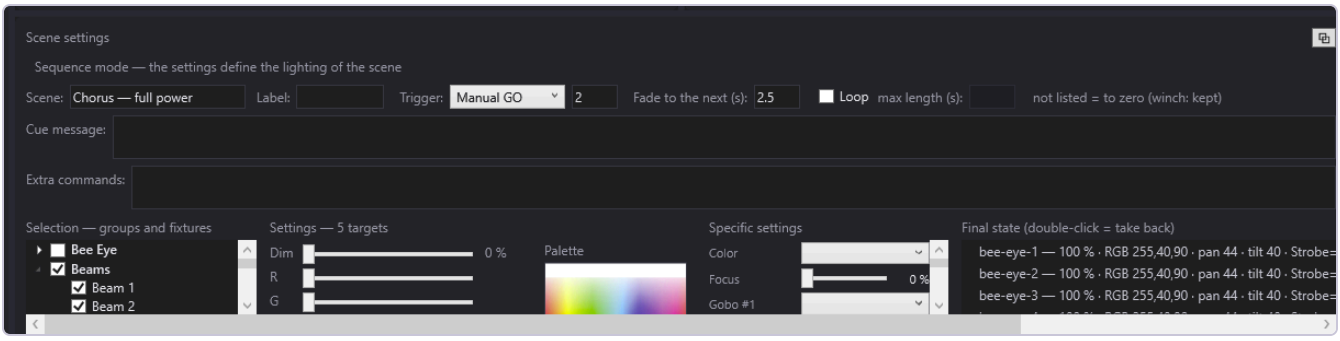
## See also

- The Scenes panel
- Selecting fixtures and adjusting the light
- Fades and mixes
- Jumps and scene labels
- Parallel sequences
- Waits (key, time, date, sun)
- Show mode and the cue window
- Sound and video on the timeline

# Selecting fixtures and adjusting the light

To light a scene, you first tick the fixtures (or groups) to adjust, then set their intensity, colour, position and special channels. These settings make up the scene's **final state**: the stable lighting it takes on when you arrive on it. Everything is done in the lower part of the **Scene settings** panel.

**To open it:** select a scene in the **Scenes** panel; the selection and the light settings are below the scene settings line, in the **Scene settings** panel.




## How a scene's light is defined

A scene's light comes from two sources:

1. **The final state**: the list of settings (intensity, colour, position, special channels) of each group or fixture for this scene. It is the stable state the fade glides towards when you arrive on the scene, and the one it keeps once its timeline has finished. Jumping straight to the scene always gives this same state.
2. **The timeline blocks**: temporary variations laid over the final state while the timeline plays (light blocks, effects, trajectories, dynamic effects, blackout…). See Light and effect blocks.

A fixture that does not appear in the final state is set to zero in this scene, unless it is set to “hold” in the venue (a winch, for example), in which case it keeps the value of the previous scene. This is the “not listed = to zero (winch: kept)” reminder shown at the top of the panel.

The settings described here write into the final state of the selected scene. Exception: in effect mode or trajectory mode (orange banner at the top of the panel), they write into the effect or the keyframe being edited.

Each setting is applied almost at once (less than a tenth of a second): the simulation shows it, and if **LIVE** is on, so do the real fixtures.

## Selection — groups and fixtures

The left-hand column, **Selection — groups and fixtures**, is a tree with check boxes:

- each **group** of the venue, in bold, with the fixtures that make it up below it (click the arrow to expand);
- then, also in bold, each fixture that does not belong to any group.

The ticking rules:

- Ticking a group ticks all its fixtures; unticking it unticks them all.
- Unticking a single fixture of a ticked group unticks the group's box, without touching the other fixtures.
- A fixture that belongs to several groups appears several times in the tree: its boxes are linked, ticking one ticks the others.
- Ticking a group never ticks another group that shares the same fixtures.

When a group **and** its fixtures are ticked, the setting is written only once, under the group's name.

### The right-click menu

A right-click in the tree (or on the 2D plan of the simulation) opens a menu:

- **Select all**: ticks all groups and all fixtures outside groups;
- **Select nothing**: unticks everything;
- for each group, **+ select** *group* and **− deselect** *group*: adds this group to or removes it from the selection, without touching the rest;
- **Import the ppshow calibration (lyrescal.ini)…**: see Moving-head calibration.

### Selecting in the simulation

You can also select from the venue view:

- a click on a fixture on the 2D plan, or on a fixture in the 3D view, ticks or unticks that fixture;
- in 2D, drawing a rectangle with the mouse ticks all the fixtures inside it (and unticks the others); with the Ctrl key, the rectangle **adds** to the selection;
- the simulation's **All** and **None** buttons do the same as **Select all** and **Select nothing**.

When all the fixtures of a group are selected this way, the group's box gets ticked, and the tree opens on the selection. See 2D and 3D simulation.

## Settings

Above the sliders, a line reminds you of the target: “Settings — no selection”, “Settings —” followed by the target's name when only one is ticked, or “Settings — 4 targets”.

The sliders offered depend on the selected fixtures:

- **Dim**: the intensity, from 0 to 100 % (the value is shown on the right). Present if a selected fixture has an intensity channel.
- **R**, **G**, **B**: the colour, in red, green and blue, from 0 to 255 each. The small rectangle to the right of **B** shows the resulting colour. Present if a selected fixture can do colour: RGB or CMY LEDs, or a colour wheel (DreamShow then picks the closest colour on the wheel).
- **Pan** and **Tilt**: the horizontal and vertical orientation of a moving head (motorised head), from 0 to 100 (50 = middle of the travel). Present if a selected fixture is a moving head.

The white channel of an RGB+white fixture has no slider: DreamShow sets it by itself to the minimum of R, G and B.

### How the sliders write

- When **only one** target is ticked and it already has a setting in the final state, the sliders move to its values: you start again from where it is.
- With several targets ticked, the sliders do not move: the value you set is given to all the targets.
- **First setting of a target**: if the target has no entry in the final state yet, touching one of the **Dim**, **R**, **G**, **B**, **Pan**, **Tilt** sliders (or clicking in the **Palette**) gives it **all** the displayed values (Dim, R, G, B, and Pan/Tilt for a moving head). So check the sliders before touching a fresh target.
- After that, each slider only writes its own value: moving the colour does not touch the position, and vice versa.
- A moving head that is set also gets its shutter open, and its pan/tilt speed at the fastest if it has one, so that its beam is visible where you send it.
- When you set the intensity or colour of a group, the fixtures of the group that had their own setting (created for example with **Aim**) get the same intensity or the same colour; their position, however, is not changed.
- Moving the colour by hand detaches the target from its colour palette; moving **Pan** or **Tilt** detaches it from its position palette (see Palettes).

### Removing from the final state

- **Remove from the final state**: deletes the settings of the ticked targets from the final state. These fixtures go back to zero in the scene (or to hold for a winch).
- **Remove the absent ones**: deletes from the final state the settings of fixtures or groups that do not exist in the current venue (leftovers from another venue). If there are none, the status line at the top of the screen briefly shows “Final state: no absent fixture”.

## Palette

To the right of the sliders, the **Palette** is a colour picture (visible when the selection can do colour):

- click a colour, or drag the mouse while holding the button down: the **R**, **G**, **B** sliders take the colour under the pointer, and the selection changes live;
- if **Cell** or **Ring** boxes are ticked in **Specific settings**, the palette colours **only** those cells (see below);
- with no box ticked, the palette colours the whole fixture and clears the individual colours of the cells.

The **Palettes…** button below opens the named palettes (colours, positions, beams): see Palettes.

The palette picture is the `colorpalette.png` file in the data folder. If it is missing, the area stays empty.

## Specific settings

The **Specific settings** column lists the other channels of the selected fixtures: shutter, colour wheel, gobo, prism, zoom, macros, speed, position of a winch… The heading only appears if there are such channels. “Fine” channels (second byte of a 16-bit channel) and reserved channels are not listed.

Each channel takes a suitable form:

- **Channel with named ranges** (at least two ranges in the fixture definition): a drop-down list that offers the ranges by their **name** (“Ouvert”, “Strobe lent à rapide”, “Rouge”…). The tooltip of each range gives its DMX values. Choosing a range sets the channel to the middle of the range.
    - If the chosen range is a real span (a strobe speed, a continuous rotation…), an extra line appears below it with a slider running from the start to the end of the range, and the DMX value on the right: set the exact value there. This slider does not appear for a “None” range or for an open shutter.
- **Continuous channel** (no ranges): a slider from 0 to 255. The value shown on the right is a percentage, or an **angle in degrees** for the zoom when the fixture definition gives the aperture of its optics. The slider's tooltip gives the DMX value and the name of the matching range.
- **Multi-cell fixture** (several colour modules, whose channels are named for example Red 1, Green 1, Blue 1, Red 2…): one **Cell N** line per module, with a check box and a rectangle showing the cell's current colour (tooltip “Tick, then click the palette to colour this cell”).
- **Light ring** (accent ring on some moving heads): a **Ring** line, with a box and a colour rectangle (tooltip “Tick, then click the palette to colour the ring”). A possible second intensity channel specific to the ring appears as a separate line.

Like the sliders, each specific setting is written for all the ticked targets, and a group passes the value on to those fixtures of the group that already have their own special channels in the final state. Changing a beam channel (zoom, focus, gobo…) detaches the target from its beam palette if that palette sets this channel.

## Final state (double-click = take back)

The right-hand column, **Final state (double-click = take back)**, lists the scene's settings, one line per group or fixture:

```
► Lyres — 80 % · RVB 255,120,0 · pan 50 · tilt 35 · Shutter=255 Zoom=128 · [Ambre, Face]
```

- ► at the start of the line: this target is currently ticked.
- The name of the group or fixture.
- The intensity in %, the colour (“RGB r,g,b”), pan and tilt (from 0 to 100), then the special channels in the form `nom=valeur`.
- In square brackets, the palettes used (colour, position, beam).

A **double-click** on a line takes that setting back: the line's target becomes the only one ticked, and the sliders move to its values.

If the final state contains settings for fixtures absent from the current venue (a show that came from another venue), a last line counts them: “(N setting(s) for fixtures absent from this venue — kept in the show, not played)”. They stay in the show file and will come back if you reopen the show in its original venue; **Remove the absent ones** deletes them permanently.

In effect mode, the heading becomes “Contents of effect “…””; in trajectory mode, “Key frame of “…””.

## Creating a group from the selection

The **+ group (selection)** button is in the **Libraries** panel, below the **Groups** list:

1. Tick the fixtures of the future group (in the tree or in the simulation).
2. Click **+ group (selection)**.
3. Give a name (“Group name (N fixtures):”) and confirm with **OK**.
4. The group is added to the venue and saved at once in the venue file. It appears in the **Selection — groups and fixtures** tree and in the **Groups** list.

The name must be neither that of an existing group nor the identifier of a fixture (“That name already exists.”). With no fixture ticked, DreamShow replies “Select the fixtures of the future group first.”.

## Step by step

### Setting a front light to amber at 70 %

1. Select the scene in **Scenes**.
2. Tick the front light group in **Selection — groups and fixtures**.

3. Set **Dim** to 70 %.
4. Click a soft orange in the **Palette**, or set **R** 255, **G** 150, **B** 40.
5. The group's line appears in **Final state**; the simulation shows the front light on.

### Putting a moving head on an actor

1. Tick the moving head.
2. Set **Pan** and **Tilt**, or more simply use **Aim** in the simulation (see Aiming and covering a zone with moving heads).
3. In **Specific settings**, choose a gobo from its list and set the zoom in degrees.

### Taking back and correcting an existing setting

1. Double-click the line you want in **Final state**.
2. Only that target is ticked and the sliders show its values.
3. Make your corrections; the line updates.

### Colouring each cell of an LED bar differently

1. Tick the bar.
2. In **Specific settings**, tick only **Cell 1**, then click the red in the **Palette**.
3. Untick **Cell 1**, tick **Cell 2**, click the blue, and so on.
4. To go back to a single colour, untick all the cells and click in the **Palette**: all the cells take that colour.

## Good to know

- The selection is shared by the whole screen: the simulation, **Aim**, **Zone**, **Calibrate...**, **+ effect (selection)**, **+ trajectory (selection)** and the palettes act on the ticked targets.
- A fixture marked “out of order” or “invisible” in the venue does not appear in the simulation, but stays in the selection tree.
- A scene's settings are only saved to disk with **File > Save** (Ctrl+S). Groups, on the other hand, are saved straight away in the venue.
- For a moving head, the simulation's **zoom on aim** box decides whether **Aim** also changes the zoom.

## See also

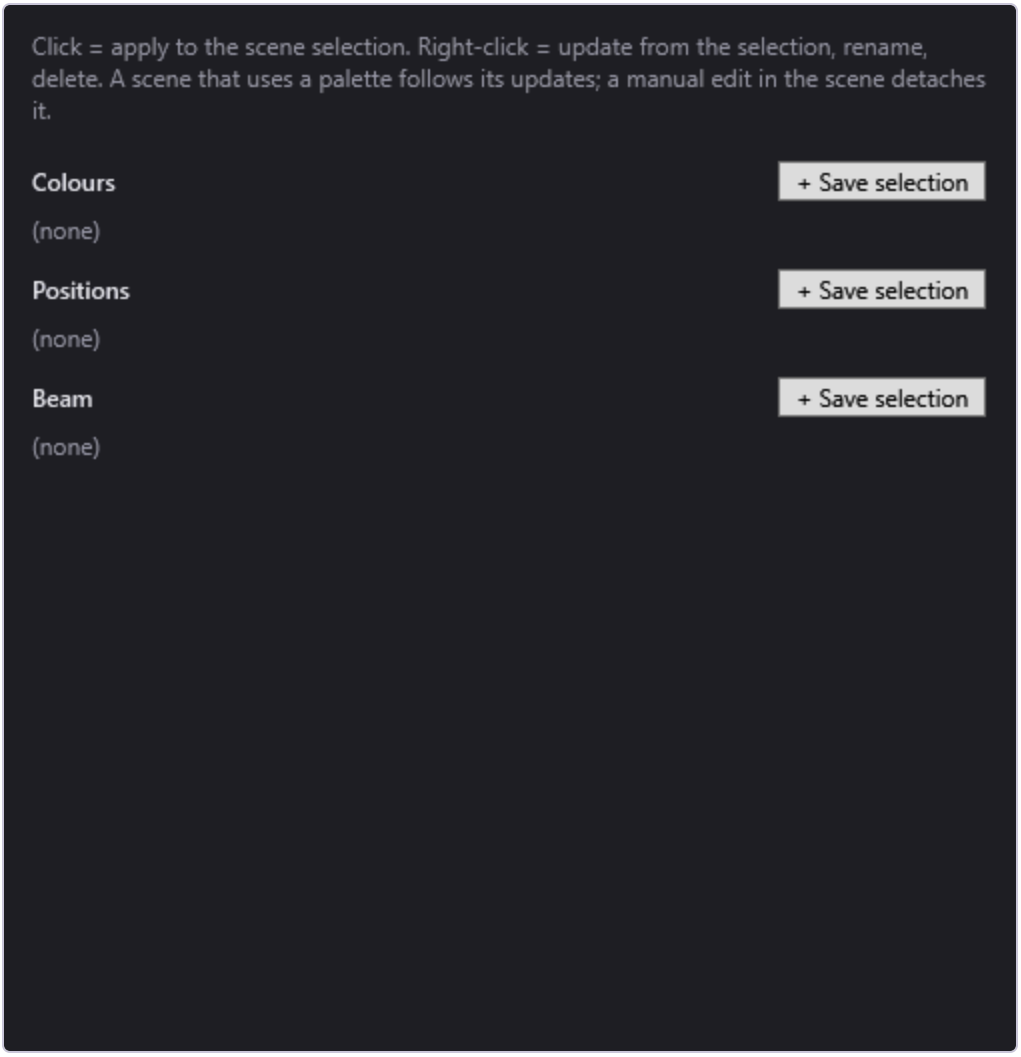
- Scene settings
- Palettes
- 2D and 3D simulation
- Aiming and covering a zone with moving heads
- Light and effect blocks

- Libraries
- Fixtures, addresses, positions

# Palettes

A palette is a named setting that you reuse in every scene: a **colour** (“Amber”, “Midnight blue”), a set of moving-head **positions** (“Front”, “Singer”) or a **beam** setting (zoom, focus, gobo, prism…). One click applies the palette to the selected fixtures. Above all, scenes keep the link with the palette: if you update it (a moving head has been moved, the tint no longer suits), every scene that uses it changes in one go.

**To open it:** the **Palettes…** button below the colour palette, in the **Scene settings** panel. This button only appears if the selection contains a fixture that can do colour. The **Palettes** window is not modal: it can stay open while you program the scenes.




## The Palettes window

At the top, a reminder: “Click = apply to the scene selection. Right-click = update from the selection, rename, delete. A scene that uses a palette follows its updates; a manual edit in the scene detaches it.”

At the bottom, a status line shows the result of the last action (palette applied, saved, error…).

In the middle, three sections. Each has a **+ Save selection** button to the right of its heading, and the list of its palettes as buttons (“(none)” when the section is empty).

### Colours

- What is saved: the colour (R, G, B) of the selection. Button tooltip: “The colour of the selection (R, G, B sliders or palette).”
- DreamShow takes the colour the first selected fixture has in the scene; failing that, the colour shown by the **R**, **G**, **B** sliders.
- The palette's button shows a strip of the colour, then the name.

### Positions

- What is saved: the pan and tilt of **each** selected moving head, as it is in the scene. Tooltip: “The pan/tilt of each selected fixture: each one returns to its own position.”
- Applied later, the palette sends each moving head back to **its** saved position. A moving head that was not in the selection when the palette was saved gets the average of the saved positions.
- The button shows the number of fixtures (“4 fixture(s)”), then the name.
- If no selected moving head has a position in the scene: “None of the selected fixtures has a position (pan/tilt) in this scene.”

### Beam

- What is saved: the beam channels of the selection, as they are set in the scene: zoom, focus, iris, frost, gobo, prism. Tooltip: “Zoom, focus, iris, frost, gobo and prism set in “Specific settings”.”
- The button shows the names of the first three channels, then the name of the palette.
- If the selection has none of these channels set in the scene: “No beam settings (zoom, focus, gobo, prism, frost, iris) on the selection in this scene: set them first in “Specific settings”.”

## The actions

### Saving a palette

1. Select the fixtures and set them in the scene.
2. Click **+ Save selection** in the right section.

3. A small window suggests a name (“Colour 3”, “Position 2”, “Beam 1”…). Change it if you like, then **OK**.
4. The palette appears in the section. Message: “Palette “…” saved. Apply it to the selection so that the scene follows it.”

Caution: saving a palette does not yet link the scene to it. Then click the palette to apply it, if you want the scene to follow its updates.

If a palette of the same type already has this name, it is replaced.

### Applying a palette (click)

A click on a palette applies it to the ticked targets of the scene (or of the effect, or of the keyframe being edited):

- the target takes the palette's values, **and** keeps a reference to the palette;
- a target with no setting in the scene is created, with the intensity of the **Dim** slider;
- a position palette only applies to targets that have pan/tilt;
- applying a colour clears the individual colours of the cells;
- when you apply a palette to a group, the fixtures of that group that had their own setting for the same attribute get the palette too.

Message: ““…” applied to the selection.” With no selection: “First select fixtures or a group in the scene.”

The palettes used by a target are recalled in square brackets at the end of its line in **Final state**.

A palette's tooltip says how many settings use it: “Used N time(s) in this show and the effects”.

### Updating a palette (right-click)

Right-click the palette, then **Update from selection**:

- the palette takes the current values of the selection (same rule as when saving it);
- every scene of the open show and every effect that uses this palette takes the new values at once;
- message: ““…” updated: N scene setting(s) follow.”

### Renaming a palette (right-click)

**Rename…**: give the new name. The references in the open show and in the effects follow the new name. A name already used in the same section is refused (“A palette with this name already exists.”).

### Deleting a palette (right-click)

**Delete**: after confirmation, the palette disappears. The scene settings that used it keep their current values, but no longer follow anything ("The N scene setting(s) that use it keep their current values.").

## Detaching a scene from its palette

A setting made by hand in the scene detaches the target from the palette concerned, and only from that one:

- moving **R**, **G**, **B** or clicking in the **Palette** picture detaches the colour palette;
- moving **Pan** or **Tilt**, using **Aim**, drawing a **Zone** or the **Calibrate…** wizard detaches the position palette;
- changing, in **Specific settings**, a channel that the beam palette sets detaches the beam palette.

The target keeps its values; it simply no longer follows the palette's updates. To attach it again, apply the palette again.

## Step by step

### A "Singer" position for four moving heads, reused everywhere

1. Select a scene, tick the four moving heads.
2. Aim at the singer (the simulation's **Aim** button, or the **Pan**/**Tilt** sliders).
3. **Palettes…** then, in **Positions**, **+ Save selection**; name it "Singer".
4. Click **Singer** to link this scene to the palette.
5. In every other scene where the moving heads must aim at the singer: select the scene, tick the moving heads, click **Singer**.
6. The day the mic stand moves: aim at the new spot in any scene, then right-click **Singer** > **Update from selection**. All the linked scenes follow.

### Correcting a tint throughout the show

1. Tick a fixture that uses the "Amber" palette in a scene.
2. Adjust **R**, **G**, **B** (this setting temporarily detaches the target).
3. Right-click **Amber** > **Update from selection**: the palette takes the new tint, and so do all the linked scenes.
4. Click **Amber** again to reattach the target you have just adjusted.

## Good to know

- Palettes are saved in the `palettes.json` file in the data folder: they are shared by all the venues and all the shows in that folder.

- A position palette saves positions per **fixture**: it makes sense in the venue where it was made.
- A scene always keeps the concrete values of its palettes: if a palette is missing (deleted, or file absent on another PC), the scene plays as before.
- **Rename…** only updates the open show and the effects. In the other saved shows, the settings that pointed to the old name keep their values but no longer follow the palette.
- Updating a palette also changes the library effects that use it; the effects file is then saved again.
- The **Palettes…** button is hidden when no fixture in the selection can do colour (for example a selection of moving heads without colour): select at least one colour fixture to open it, then keep the window open.

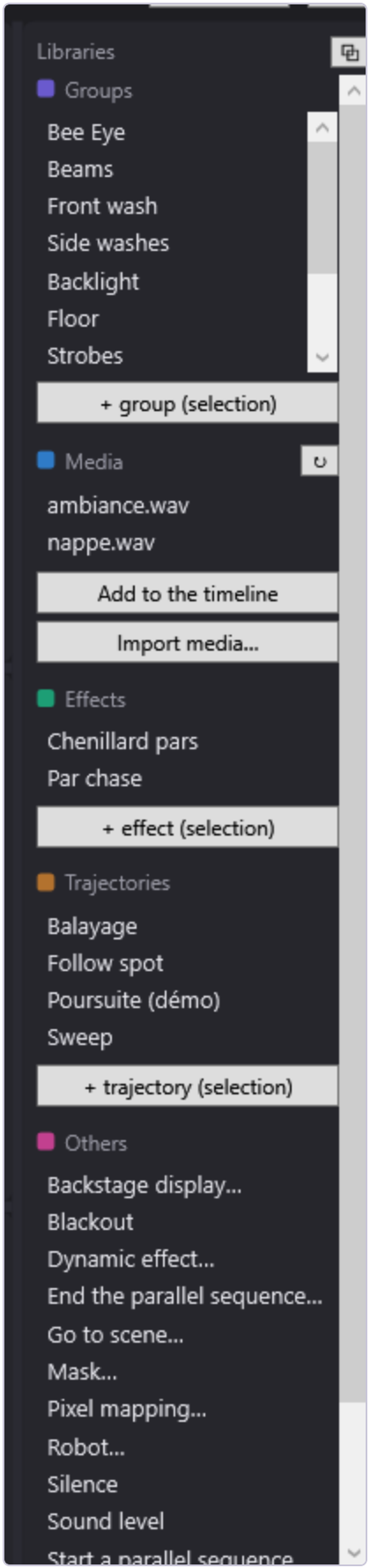
## See also

- Selecting fixtures and adjusting the light
- Aiming and covering a zone with moving heads
- Libraries
- Light and effect blocks

# Libraries

The **Libraries** panel brings together everything you place on a scene's timeline: fixture groups, media (sounds, videos, pictures, DMX streams), effects, trajectories and the other events (jumps, waits, blackout, volume…). You drag them onto a timeline track; some can also be created directly here, from the selection.

**To open it:** the panel is in the right-hand column of the main screen (default layout).




The panel has five sections. In front of each heading, a small coloured square recalls the colour of the matching blocks on the timeline (the square's tooltip says so).

| Section | Block colour | Square's tooltip |
|---|---|---|
| **Groups** | purple | Colour of light blocks on the timeline |
| **Media** | blue | Colour of media blocks on the timeline |
| **Effects** | green | Colour of effect blocks on the timeline |
| **Trajectories** | orange | Colour of trajectory blocks on the timeline |
| **Others** | pink | Colour of “Other” events on the timeline |

## Dragging an item onto the timeline

The gesture is the same for all the lists:

1. Click an item in the list and, without releasing, move the mouse a few pixels.
2. Bring it onto a track of the **Scene timeline**, at the place (time) you want.
3. Release: the block is created there.

The details of the blocks (duration, fades, properties) are in The scene timeline and the pages that follow.

## Groups

The list of the venue's groups.

- **Dragging a group** onto a track creates a 10-second **light block** for that group. The block takes the group's setting from the final state of the selected scene; failing that, its setting in another scene of the show; failing that, white at 100 %. See Light and effect blocks.
- **+ group (selection)** (tooltip “Creates a group with the selected fixtures”): creates a group with the ticked fixtures. DreamShow asks for its name (“Group name (N fixtures):”). The group is saved at once in the venue and appears in the selection tree. A name already used by a group or by a fixture is refused. See Selecting fixtures and adjusting the light.
- **Right-click > Delete the group**: removes the group from the venue, after confirmation. The fixtures stay; only the show settings that targeted this group no longer target anything.

Groups can also be managed in **Venue > Fixtures, addresses, positions...** (see Fixtures, addresses, positions).

## Media

The list of the files in the show's media folder: `medias\<nom du show>` in the data folder (the folder opens with **File > Open the show's media folder**). Each show has its own media folder. All the files in the folder are listed, in alphabetical order.

The list updates by itself when a file is added, deleted or renamed in the folder (for example from Windows Explorer).

- ↻ (tooltip “Refresh the media list”): rereads the folder by hand. Timeline blocks whose file is missing are flagged again, and the video player rereads its folder.
- **Add to the timeline**: adds the selected media to the scene, at time 0.
  - The block goes on the track chosen in the timeline; failing that, on a track named “Media” (created if needed).
  - Its duration is the real duration of the file (rounded up to the next tenth of a second), or 30 seconds if it is unknown; DreamShow refines it as soon as it has been able to open the file.
  - A DMX stream file ( `.DmxStreaming` , `.bin` , `.dmx` ) becomes a DMX block, on the chosen track or on a “DMX” track, with the real duration of the recording.
  - If no media is selected, a file-open window appears; the chosen file is **copied** into the show's media folder (if it is not already there), then added.
- **Import media…** (tooltip “Copies one or more files into the show's media folder”): choose one or more files; they are copied into the show's media folder. A file with the same name already there is **replaced**. The status line at the top of the screen confirms “N media imported into medias\…”.
  - The types offered: “Media” ( `.mp4` , `.avi` , `.mov` , `.mkv` , `.mp3` , `.wav` , `.flac` ), “DMX streaming” ( `.DmxStreaming` , `.bin` , `.dmx` ) and “All”.
- **Dragging a media file** onto a track: creates a media block (or a DMX block for a DMX stream) at the time where you drop it.

### The right-click menu on a media file

Right-clicking selects the media under the pointer and opens:

- ▶ **Play**: listen to or watch the media straight away, to try it out. A sound is played by DreamShow; a video is sent to the ppvideoplayer video player (started if needed).
- ■ **Stop**: stops all sounds, and the selected video.
- **Edit with GoldWave**: opens the file in the GoldWave audio editor, if it is installed. DreamShow looks for it in the usual installation folders; if it cannot find it, it asks where `GoldWave.exe` is and remembers it.
- **Open the media folder**: opens Windows Explorer on the media folder, with the file selected.
- **Delete the media**: erases the file from the media folder, after confirmation. If the media is used on a timeline, the confirmation warns: “WARNING: this media is used on the timeline — the blocks will turn into errors.”

See Sound and video on the timeline.

## Effects

An effect is a named light setting (groups or fixtures and their values), reusable in every scene and every show. Effects are saved in the `effects.json` file in the data folder.

- **Dragging an effect** onto a track creates a 10-second **effect block**. The block refers to the effect by its name: changing the effect changes every block that uses it.
- **+ effect (selection)** (tooltip “Saves the selection and its settings as a named effect (double-click = edit an effect)”):
    1. Tick groups or fixtures and make your settings.
    2. Click **+ effect (selection)** and give a name (“Effect name:”).
    3. The effect saves, for each ticked target, the values of the **Dim**, **R**, **G**, **B** sliders (and **Pan**, **Tilt** for moving heads), plus the special channels the target has in the scene's final state.

With no target ticked: “Tick groups/fixtures and make your settings, then create the effect.” A name already in use is refused.

- **Double-click** an effect, or **right-click > Edit the effect**: switches to **effect mode**. The banner of the **Scene settings** panel turns orange (“EFFECT MODE — “…”: the settings change the effect, not the sequence”), the effect's targets are ticked, and all the settings (sliders, palette, specific settings, **Aim**…) change the effect. The simulation shows the effect over the selected scene. The right-hand list is headed “Contents of effect “…””.
- **✓ Finish editing the effect** (on the right of the banner): saves the effects and goes back to sequence mode.
- **Right-click > Delete the effect**: deletes the effect, after confirmation.

See Light and effect blocks.

## Trajectories

A trajectory is a moving-head movement described by keyframes (positions, colours, zones) over time. Each trajectory is a file in the `trajectoires` folder of the data folder.

- Tooltip of the list: “Double-click = open the key-frame editor; drag onto a track = insert; Del = delete”.
- **Dragging a trajectory** onto a track creates a **trajectory block** as long as the trajectory.
- **Double-click**: opens the keyframe editor and switches to trajectory mode.
- **Delete** (key): deletes the selected trajectory, after confirmation.
- The **right-click** offers:
    - **Open the editor**;
    - **Duplicate…**: saves a copy under another name (suggested: the name followed by “(copy)”); the original and its blocks are not touched;
    - **Delete…**: erases the file permanently, after confirmation. Blocks already placed on a timeline remain, but no longer play anything.

- **+ trajectory (selection)** (tooltip “Creates a trajectory for the ticked heads: their current state becomes the first key frame, then the editor opens”): first tick and position the moving heads, click, give a name; the editor opens.

All the details are in Trajectories.

## Others

Events to drag onto a track (tooltip of the list: “Drag onto a track: instant events”). The list, sorted in alphabetical order (of the French names), contains:

| Item | In brief | Detailed page |
|---|---|---|
| **Backstage display…** | sends a message to the backstage monitors | Backstage displays |
| **Go to scene…** | jumps to another scene (DreamShow asks which one when you drop it) | Jumps and scene labels |
| **Wait for a time, a date, the sun…** | stops the show until a duration, a time, a date or sunrise/sunset | Waits |
| **Wait for a key…** | stops the show until a key is pressed | Waits |
| **Start a parallel sequence…** | starts a second playback from a scene | Parallel sequences |
| **Dynamic effect…** | a waveform applied to an attribute of a group of fixtures (wave, chase, fan of circles…) | Dynamic effects |
| **Mask…** | controls an LED mask | LED masks |
| **Sound level** | changes the left/right volume | Sound and video on the timeline |
| **Blackout** | forces all intensities to zero for the length of the block (positions unchanged) | Light and effect blocks |
| **Pixel mapping…** | plays content (pattern or picture) on the cells of bars, matrices and fixtures seen as a picture | Pixel mapping |
| **Robot…** | controls a humanoid robot | Humanoid robots |
| **Silence** | fades out all sounds, over the length of the block (a zero-width block cuts dead) | Sound and video on the timeline |
| **Stop sound** | stops the sounds playing (all of them, or those of certain tracks) | Sound and video on the timeline |

| | | |
|---|---|---|
| **End the parallel sequence…** | stops a parallel sequence with a fade | Parallel sequences |

For most items, a settings window opens when you drop them, because the block makes no sense without it: **Backstage display…**, **Go to scene…**, the two waits, the two parallel sequences, **Dynamic effect…**, **Pixel mapping…**, **Robot…** and **Mask…**. Cancelling this window adds nothing. **Sound level**, **Blackout**, **Silence** and **Stop sound** are placed directly; you then set them on the timeline.

## Step by step

### Putting music on a scene

1. Click **Import media…** and choose the `.mp3` or `.wav` file.
2. The file appears in **Media**.
3. Select the scene, then drag the file onto a track at time 0 (or select it and click **Add to the timeline**).
4. The blue block takes the length of the track.

### Creating and reusing a “Blue mood” effect

1. Tick the groups concerned, set **Dim** and the colour.
2. Click **+ effect (selection)**, name it “Blue mood”.
3. Drag “Blue mood” onto the timeline of any scene, of any show.
4. To retouch it everywhere: double-click it, adjust, then **✓ Finish editing the effect**.

## Good to know

- Media are referenced by their file **name** only, in the show's media folder: copying the project to another PC (**File > Export the whole project (zip)…**) keeps everything in place.
- **Import media…** replaces without warning a file that already has the same name in the folder.
- Effects and trajectories are shared by all the shows in the data folder; media belong to a show.
- Changes to an effect are saved in `effects.json` when you click **✓ Finish editing the effect**.
- **Delete the media** and **Delete…** (trajectory) really erase the files from the disk.

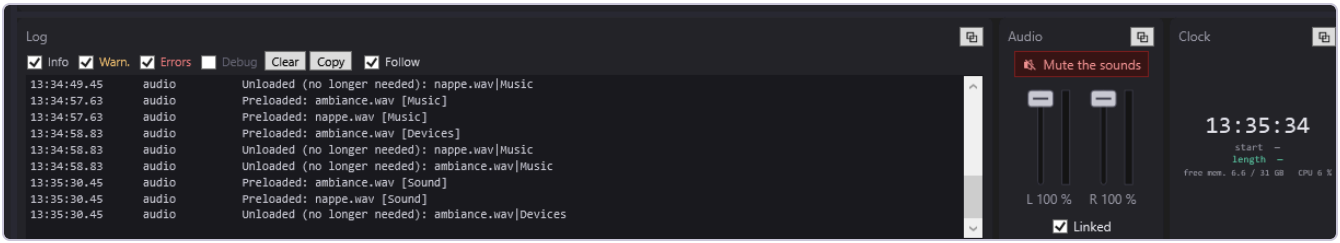
## See also

- The scene timeline
- Light and effect blocks
- Sound and video on the timeline
- Trajectories
- Selecting fixtures and adjusting the light
- Shows, files and projects

# Log, Audio and Clock

Three small panels take up the bottom of the main screen. The **Log** tells you what DreamShow is doing (scenes played, sounds started, interface errors…). The **Audio** panel gives you an emergency button to cut the sounds, the left and right volumes, and level meters. The **Clock** shows the time, the start time and length of the show, and the state of the PC.

**To open them:** the three panels are at the bottom of the main screen, from left to right: **Log**, **Audio**, **Clock** (default layout). Like all panels, they can be detached into a floating window with ⧉ (see Main screen overview).




## The Log panel

### The log lines

Each line reads like this:

```
20:41:07.52 AVT dmx Connexions : …
```

- the time, to the hundredth of a second;
- the level of the message: nothing for information, “WRN” for a warning, “ERR” for an error, “dbg” for a diagnostic message;
- the source: the part of DreamShow that is speaking (show, dmx, audio, video, venue, extra, robot…);
- the message.

The colours help you spot things: light grey for information, orange-yellow for warnings, red for errors, dark grey for diagnostics.

You can read there, for example: DreamShow starting up, each Go (“GO → scene 3 (Storm), fade 2 s”), automatic follow-ons, sounds and videos started and stopped (with their screen), the extra commands executed, the show being saved, the end of the show (“END OF SHOW: last scene reached — length …”), and DMX interfaces that could not be opened.

### The filters and buttons

At the top of the panel:

- **Info** (ticked by default): shows information messages.
- **Warn.** (ticked by default): shows warnings.
- **Errors** (ticked by default): shows errors.
- **Debug** (unticked by default): shows diagnostic messages (recompilation times, simulation rate, aiming details…). Useful for tracking down slowness or for support.
- **Clear**: empties the log.
- **Copy** (tooltip “Copies the displayed journal to the clipboard”): copies to the clipboard all the **displayed** lines (according to the filters), so you can paste them into an e-mail or a help request.
- **Follow** (ticked by default): the log scrolls by itself to show the last line. Untick it to read an older message in peace while others keep arriving.

The filters also act on messages already received: ticking **Debug** brings up the diagnostic messages of the previous minutes.

### Good to know about the log

- The log keeps the last 1,000 messages. The oldest ones disappear.
- It is not saved to disk: it is emptied when DreamShow closes. To keep a record (for example after an incident), use **Copy** before quitting.
- Messages issued during loading, before the window opens, are included too.

## The Audio panel

The **Audio** panel controls the sounds played by DreamShow itself (music, sound effects from media blocks).

- **Mute the sounds** (red button; tooltip “Immediately stops ALL playing sounds (editor as well as showtime).”): stops dead all the sounds DreamShow is playing, in editing as well as during a performance. The cue window has a button that does the same.
- **Left volume** and **right volume**: two vertical faders, from 0 to 100 %. Up = louder. The value is shown below: “L 100 %” and “R 100 %”. A click on the track brings the fader straight there.
- **Level meters**: to the right of each fader, a green bar shows the level of the sound DreamShow is sending, left side and right side. It measures the sound **before** the faders: it moves even if a fader is down. It does not show the sound of other programs (nor that of the video player).
- **Linked** (box below the faders): when ticked, the two faders move together; at the moment you tick it, the right fader lines up with the left one.

The volumes and the **Linked** box are remembered and restored the next time DreamShow starts.

A **Sound level** event placed on the timeline changes the volume during the show: the panel's faders then move with it. See Sound and video on the timeline.

### Good to know about audio

- **Mute the sounds** and the faders concern the sounds played by DreamShow. Videos are played by the ppvideoplayer video player, which is not stopped by this button.
- The Windows master volume and that of the mixing desk add to these faders.

## The Clock panel

The **Clock** panel grows with the space it is given: its content is scaled to fill the panel, which lets you turn it into a big control-room clock (by detaching it with ⧉ onto a separate screen, for example).

- **The time**: the PC's time, in large type (hh:mm:ss).
- **start**: the start time of the show, that is of the first Go given from the “Ready” state (before the first scene, for example when the cue window opens). “start —” as long as the show has not begun.
- **length** (in green): the time elapsed since the start, in h:mm:ss. When the show reaches its **last scene**, the length stops and “END hh:mm:ss” appears next to it: this is the time the show ended. “length —” as long as the show has not begun.
- **free mem. … / … GB CPU … %**: the free memory and total memory of the PC, and the processor load of the whole PC (whereas the status line at the top of the screen gives DreamShow's share alone).

The clock is reset (“—”) when the show goes back to the “Ready” state, for example when you open the cue window for a fresh start.

## Step by step

### Sending the log to someone who is helping you

1. In the **Log**, tick **Debug** if you have been asked to.
2. Reproduce the problem.
3. Click **Copy**.
4. Paste (Ctrl+V) into your message.

### Turning the music down by half during a rehearsal

1. Check that **Linked** is ticked.
2. Move one of the faders down to “L 50 %”: the other follows.

### Showing a large clock at the control position

1. Click ⧉ at the top right of the **Clock** panel.
2. Place the window on the screen you want and maximise it: the time, the start and the length grow with it.

## See also

- Main screen overview
- Sound and video on the timeline
- Show mode and the cue window
- Troubleshooting

# 2D and 3D simulation

The **Venue** panel shows the venue and its fixtures, seen from above (2D) or in volume (3D), with their beams, their colours and the video screens. It lets you see the show without the equipment, select fixtures with a click and point moving heads with the mouse. You can write and rehearse the whole show without any fixture connected.

**To open it:** the panel, titled **Room — click = select**, is at the top centre of the main screen, to the left of the timeline (default layout). It can be detached into a floating window with ⧉, for example to show it large on a second screen.




## The toolbar

From left to right:

| Item | Role |
|---|---|
| ⏻ | turns the simulation rendering on or off |
| **2D** / **3D** | chooses the view |
| **Basic** / **Standard** / **Maximal** | quality of the beams in 3D (visible in 3D only) |

| | |
|---|---|
| **All** / **None** | selects all the fixtures / none |
| **Aim** | a click in the venue points the selected moving heads |
| **Zone** | draws a zone that the selected moving heads share between them |
| ◯ | elliptical zone (unticked = rectangle) |
| **zoom on aim** | **Aim** also sets the zoom for a spot about 1 m wide |
| **Rot°** | rotation of the zone, in degrees |
| **Calibrate…** | 4-point calibration of the selected moving head |
| **Expo** | exposure of the simulation, in lux |
| **Show** ▾ | visible fixture categories, background plan |

### ⏻ — simulation rendering

Button on by default. Tooltip: “Simulation rendering. Off: the room is no longer computed or drawn (2D and 3D), which relieves the processor and the graphics card during a show. The DMX output is not affected.”

- Off: the panel shows “Simulation off — the ⏻ button turns it back on. The DMX output carries on normally.” The videos on the simulation's screens are paused. The log notes “Simulation off (the DMX output carries on)”.
- Back on: the venue is redrawn at once (in 3D, the view is reframed on the whole venue).
- This choice is **remembered**: if you quit DreamShow with the simulation off, it stays off the next time DreamShow starts.

### 2D / 3D

- **2D**: the plan of the venue seen from above. This is the view at startup.
- **3D**: the venue in volume (floor, stage, back wall), the fixtures, the beams in the air and their patches on the surfaces. Each time you switch to 3D, the view is reframed: you look at the stage from the audience, with the whole venue in the field of view.

The 2D / 3D choice is not remembered from one session to the next.

### Basic / Standard / Maximal — 3D quality

Visible only in 3D. Tooltip: “3D beam rendering quality: Basic (flat cones), Standard (gradient core and halo, soft impact patches), Maximal (GPU shader rendering, volumetric beams with a continuous gradient).”

- **Basic**: the lightest; choose it on a modest PC.
- **Standard**: more realistic beams, at a moderate cost.

- **Maximal**: the beams are drawn directly by the graphics card. If the graphics card does not allow it, DreamShow goes back to **Standard** and the log says so ("GPU rendering unavailable (…) — Standard quality used.").

The choice is remembered.

### All / None

- **All**: ticks all groups and all fixtures outside groups.
- **None**: unticks everything.

### Aim, Zone, ◯, zoom on aim, Rot°

These tools point the selected moving heads from the venue view. **Aim** and **Zone** are mutually exclusive: turning one on turns the other off.

- **Aim** (tooltip "Click in the room = aim the selected heads"): a click, or a drag with the button held down, points all the selected moving heads at this point; they converge. In 3D, the aimed point is the surface under the pointer: floor, stage or a wall (height included).
- **Zone**: draw a zone on the plan; the selected moving heads share it between them. The zone stays displayed and can be moved, resized and rotated with the mouse. In 2D, the plan then shows the walls unfolded around the floor (strips "back wall", "front wall", "stage right", "stage left"): a zone pushed against a wall gradually climbs up it.
- ◯: elliptical zone; unticked, the zone is a rectangle.
- **zoom on aim**: unticked (default), **Aim** only sets pan and tilt, and your zoom is kept; ticked, aiming also tightens the beam for a spot about 1 m wide on the target. The **Zone** tool, for its part, always sizes the beam. This choice is remembered.
- **Rot°**: the rotation of the zone in degrees; type a value then Enter. Values beyond 360 are allowed (720 = two turns), which is useful for trajectories.

All the details are in Aiming and covering a zone with moving heads.

### Calibrate…

Tooltip: "4-point calibration of the selected head: put the beam on each stage corner (Pan/Tilt sliders), capture; the geometric model pre-fills". Only one moving head must be selected; otherwise the status line shows "Calibrate: select a moving head" or "Calibrate: one head at a time". See Moving-head calibration.

### Expo — exposure

A slider followed by the value in lux (1000 lx by default, in the middle of the slider). Tooltip: "Simulation exposure: the illuminance (lux) that reads as "well lit"".

- The slider goes from 100 lx (left) to 10,000 lx (right), on a logarithmic scale.

- The **lower** the value, the brighter the venue looks: a weak beam already looks strong. The **higher** it is, the more light is needed for a surface to look well lit.
- Exposure only acts on fixtures whose light output is known (flux given in their definition or in the venue); the others are shown according to their colour and intensity alone.
- The value is remembered.

### Show ▾

A button that opens a small list (tooltip “Fixture categories visible in the venue”):

- **All** / **None**: shows all the categories / hides them all;
- one box per fixture category present in the venue, in the colour of the category: **“Lyres”** (moving heads), **“LEDs couleur”** (colour LEDs), **Dimmers**, **Scanners**, **Smoke**, **“Relais”** (relays), **Others**;
- **Background plan**: shows under the venue the plan picture defined for the venue (box greyed out if the venue has no plan picture).

When categories are hidden, the button becomes “Show (N hidden) ▾”. Hiding a category only changes the display: the fixtures keep receiving their DMX. These choices are remembered.

## What the 2D view shows

- The venue to scale, fitted to the size of the panel. The **stage** is a darker rectangle marked “scene”. Upstage is at the top, the auditorium (the audience) at the bottom.
- The **background plan**, if there is one and if **Background plan** is ticked.
- Each fixture is a dot in the colour it emits (grey if it is off). Hover over it: its name is shown.
- **Selected** fixtures have an orange halo.
- **The arrows**: each moving head, and each oriented fixed fixture, carries an arrow that shows the direction of its beam, whether it is on or not. The arrow turns with the beam, and gets shorter when the beam becomes vertical (a moving head aiming straight down below itself has a short arrow). It takes the colour of the light when the fixture is lit, orange when it is selected, grey otherwise.
- The **beams**: a graded cone from the source to the impact patch, on the floor or on a wall. Its width follows the real zoom when the fixture definition gives its optics. A multi-cell fixture has a cone striped with its colours. Colour LEDs and dimmers (fixed fixtures) also show their beam as soon as they are lit.
- The **light rings** of some moving heads are shown as a coloured circle around the dot.
- The **strobe** is animated, even with the timeline paused, just as the DMX output does.
- The **video screens** placed in the venue are a thick line with a small dash on the side they face, and their name. The line lights up in light blue when a video or a picture is played on it. See Video screens.

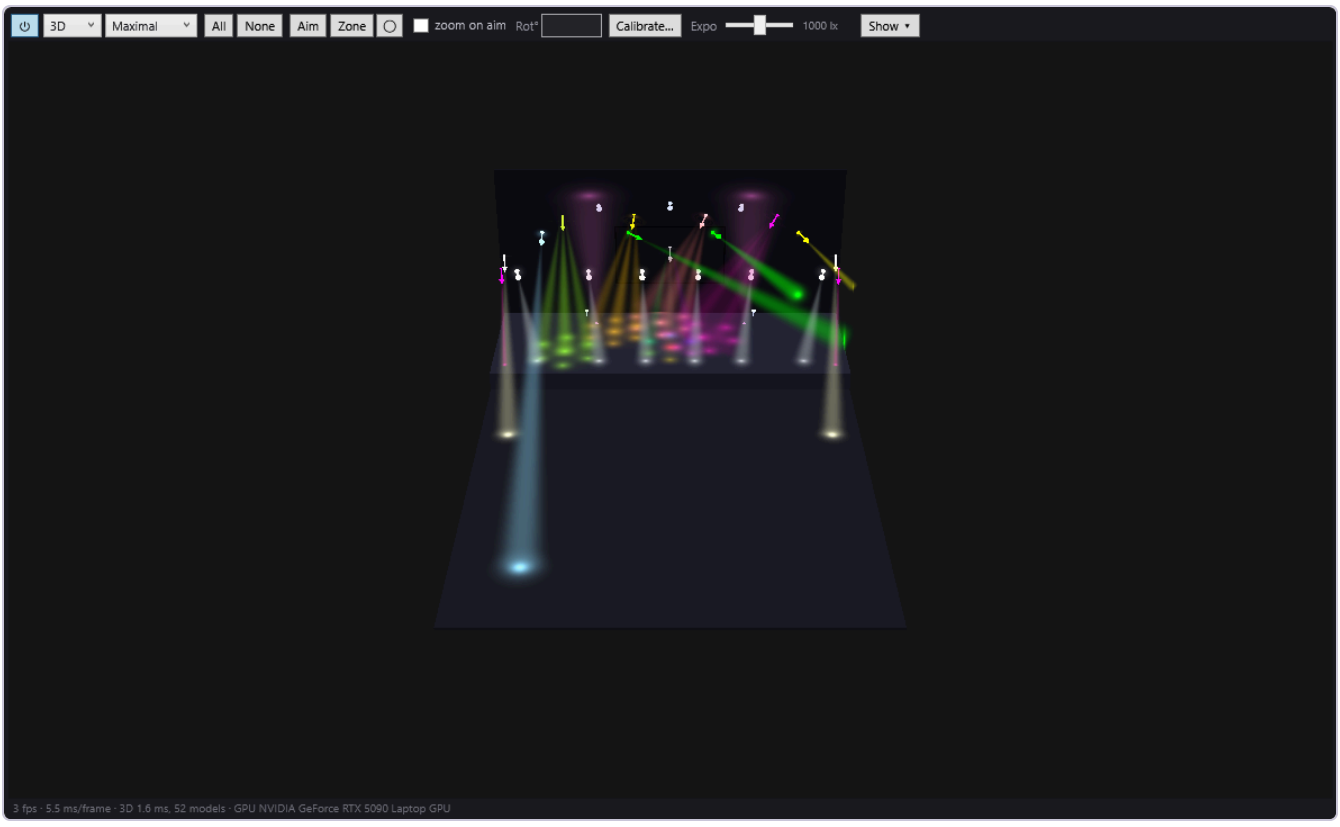
Fixtures marked “out of order” or “invisible” in the venue, and those without a position, do not appear.

### The mouse in 2D

- **Click on a fixture**: ticks or unticks it (without **Aim** or **Zone**).
- **Drag on empty space**: draws a rectangle; on release, the fixtures inside it are selected and the others deselected. **Ctrl** held down on release: the rectangle **adds** to the selection.
- **Right-click**: menu **Select all**, **Select nothing**, **＋ select** / **－ deselect** each group, **Import the ppshow calibration (lyrescal.ini)…**.
- With **Aim**: click or drag = point. With **Zone**: draw, move, resize, rotate the zone.

The 2D view has no zoom or panning: it always frames the whole venue in the panel. Enlarge the panel (or detach it) to see it bigger.

## What the 3D view shows




- The floor, the stage, the back wall.
- Each moving head and each oriented fixture is a 3D **arrow**, along the axis of its beam, visible even when off; a fixture with no direction stays a cube. Colour of an arrow: that of the light when the fixture is lit, orange if it is selected (and off), grey otherwise. A cube takes the colour of its light, and an orange halo when it is selected.
- The **beams** in the air and their patches on the floor and walls, according to the chosen quality.

- The **video screens** placed in the venue show the video or picture playing at the position of the red line, with the block's fades, without the sound (the sound belongs to the real playback chain). When paused, the picture follows the red line as you move it. See Video screens.

### The mouse in 3D

| Gesture | Effect |
|---|---|
| drag with the left button (without **Aim** or **Zone**) | orbit around the venue |
| drag with the right button | orbit around the venue, always (even with **Aim** or **Zone**) |
| wheel | move closer / further away |
| drag with the middle button, or **Shift** + drag | pan the view parallel to the screen |
| left click without moving on a fixture | ticks or unticks it |
| left click or drag with **Aim** | point the selected moving heads at the surface aimed at |
| left click or drag with **Zone** | draw or change the zone on the surface under the pointer |

- The camera always stays above the floor: you cannot go underneath.
- Zooming with the wheel is limited (between about 4 m and 400 m from the centre of the view).
- Switching to 2D then back to 3D, or turning ⏻ back on, reframes the view on the whole venue.

## The information line

Below the venue, a line in small grey type gives the smoothness of the simulation (tooltip: “Refresh rate of the room view and cost of one frame on the UI thread (3D: off-thread beam rebuild + apply, model count, GPU used at Maximal quality)”):

```
18 i/s · 6.2 ms/image · 3D 4.1 ms, 212 modèles · GPU …
```

- **i/s**: the number of frames of the venue drawn per second;
- **ms/image**: the longest time spent preparing a frame, over the last second;
- in 3D: the time taken to build the beams and the number of 3D models;
- in **Maximal**: the name of the graphics card used.

The venue is only redrawn when something changes (playback running, adjustment, strobe…): a low frame rate when nothing is happening is normal.

## What the simulation shows depending on the situation

- **The selected scene is the one the engine is playing** (normal case with **LIVE** on, in editing as well as in performance): the simulation reads the real DMX output, frame by frame. It follows the red line: fades, blocks, trajectories, dynamic effects, strobes. During playback, moving heads move at a realistic speed (limited, and slower if their pan/tilt speed channel is set slow), like the real equipment; when paused or when you move the red line, they are shown directly at their position.
- **The selected scene is not being played** (**LIVE** off, for example): the simulation shows the **final state** of the selected scene, computed as the DMX output would.
- **Effect mode** or **trajectory mode**: the simulation shows the effect (or the keyframe) **over** the scene's final state, even when paused or with **LIVE** off.
- **Visualiser mode**: the simulation shows the DMX sent on the network by another program or another device, and DreamShow no longer sends anything. See Visualiser mode.
- **Follow spot on a tablet**: moving heads driven by finger are tracked live, even with the timeline paused. See Follow spot on a tablet.
- In **Show mode**, the main screen's simulation follows the scene played by the cue list.

## Step by step

### Seeing the show in 3D

1. Choose **3D** in the list at the top of the panel.
2. Choose **Standard** (or **Maximal** if the PC can handle it).
3. Press the space bar (Go) or start timeline playback ▶.
4. Orbit around the venue by dragging with the right button; move closer with the wheel.

### Selecting all the moving heads on the stage-left side

1. Switch to **2D**.
2. Draw a rectangle around the moving heads you want: they are selected (orange halo) and ticked in the selection tree.
3. To add others, draw a second rectangle while holding **Ctrl**.

### Relieving a slow PC during the performance

1. Set the 3D quality to **Basic**, or switch to **2D**.
2. If that is not enough, turn ⏻ off: the venue is no longer drawn, the DMX output carries on normally.

### Showing only the moving heads

1. Click **Show** ▾ .

2. Click **None**, then tick **“Lyres”**.
3. The button reads “Show (N hidden) ▾”. To show everything again: **Show** ▾ then **All**.

## Good to know

- The simulation is a working tool: it approximates the real result (colours, zoom, output if known), but does not replace a test on stage.
- While you are drawing a selection rectangle or orbiting the 3D view, the simulation no longer follows playback: it resumes as soon as you release.
- Fixture categories come from the type declared in their definition (moving head, colour LED, dimmer, scanner, smoke); a model whose name starts with “Relais” is classed as **“Relais”**.
- A recent graphics card noticeably improves the 3D. On first launch, DreamShow asks Windows to use the most powerful graphics card in the PC (the “high performance” setting in the Windows graphics settings), then restarts by itself to take advantage of it.

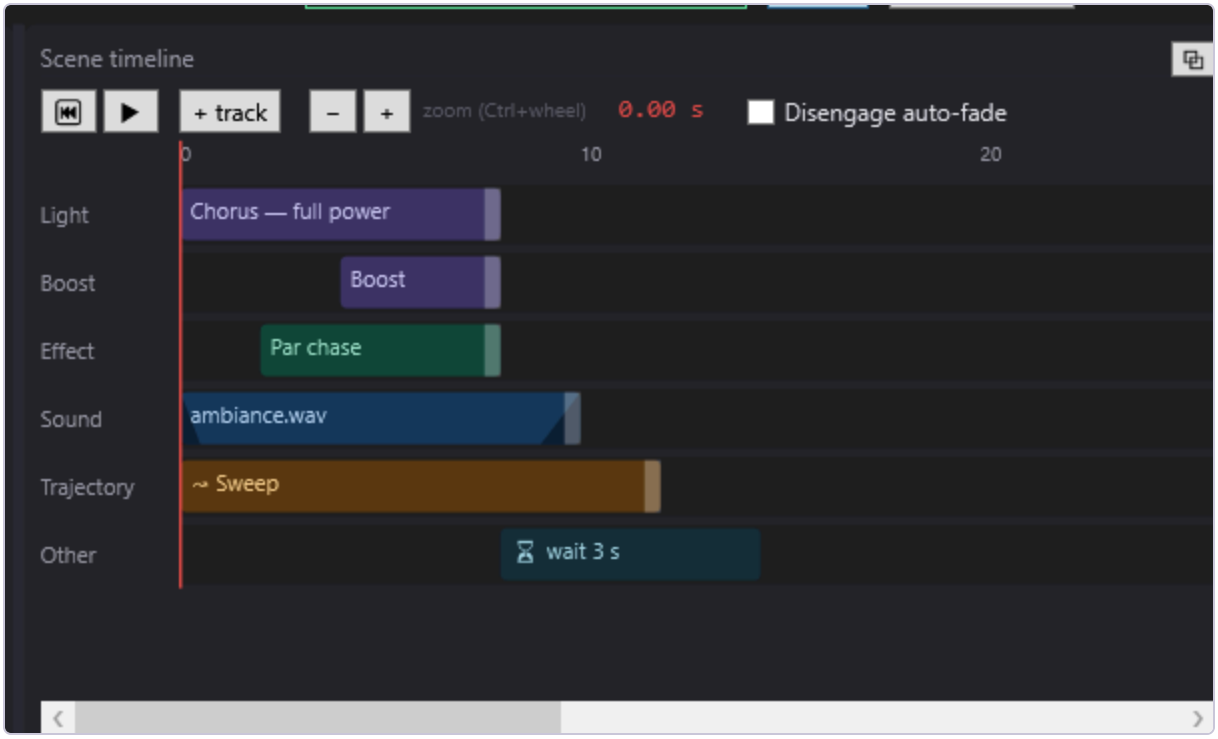
## See also

- Aiming and covering a zone with moving heads
- Moving-head calibration
- Selecting fixtures and adjusting the light
- Video screens
- Fixtures, addresses, positions
- Visualiser mode
- Main screen overview

# The scene timeline

Each scene of your show has its own timeline: a strip of time, divided into tracks, on which you drop blocks (light, effects, sounds, videos, waits, jumps…). When the scene starts, a red line runs across the timeline from left to right and triggers each block when it reaches it. When the timeline has finished, the scene settles on its **final state** (the light setting defined in **Scene settings**).

**To open it:** the timeline is the **Scene timeline** panel of the main screen (at the top, to the right of the simulation). It always shows the scene selected in the **Scenes** panel.




## What to understand first

- **One timeline per scene.** Changing scene in the **Scenes** panel shows another timeline.
- **Time zero** of the timeline is the end of the scene's incoming fade. During the fade (set by **Fade to the next (s)** of the previous scene), nothing is triggered; the timeline starts when the scene has fully arrived. See Fades and mixes.
- **A block is a temporary excursion.** A light block acts for its duration, over the final state; when it ends, the final state takes over again. See Light and effect blocks.
- **The end of the timeline** is the end of the last block, all tracks taken together.
- Any change to the timeline is taken into account at once (in less than a second), even during playback.

## The toolbar

It stays at the top of the panel; it does not scroll with the timeline.

- ⏮: “Bring the playhead back to zero”. The red line goes back to the start of the selected scene. If the engine was playing another scene, it first jumps to the selected scene. Blocks placed exactly at 0 s are played again (including the scene's extra commands).
- ▶ / ⏸: “Play / pause the timeline (editing: paused by default)”. One click starts playback (the button becomes ⏸); a second click pauses. The button follows the real state of the engine: if a GO restarts playback from the cue window, it switches to ⏸ by itself.
- **+ track**: adds an empty track at the bottom of the scene, named “Track N” (N = number of tracks + 1).
- **−** and **+**: zoom the timeline out and in (each click multiplies or divides the scale by 1.25).
- **zoom (Ctrl+wheel)**: a reminder of the shortcut. Ctrl + mouse wheel, anywhere on the timeline, zooms while keeping the moment under the pointer fixed. The wheel alone, over the graduated ruler, also zooms.
- **Playhead position** (red text, tooltip “Playhead position”): shows the time of the red line, for example “12.35 s” or “1:05.20” beyond one minute. It shows “fade 2.4 s” during the scene's incoming fade, and the text of the current wait when the scene is stopped on a wait block (see Waits). It is empty when the engine is playing a scene other than the one displayed.
- **Disengage auto-fade**: see below, dedicated section.

## The graduated ruler

At the top of the tracks, the ruler shows the seconds (then “1:00”, “1:30”… beyond one minute). The spacing of the marks adapts to the zoom.

- **Click in the ruler**: “Click = place the playhead”. The red line jumps to that moment. If the engine was playing another scene, it first jumps to the selected scene. Sounds and videos realign to the right position in the file.
- The timeline shows at least 60 seconds, and always 10 seconds more than the end of the last block.

## The red line

The red line crosses the ruler and all the tracks. It shows how far playback has got in the scene **the engine is playing**. It only appears if the scene displayed is the one the engine is playing.

- During playback, the timeline scrolls to keep the red line visible (it moves back to 20 % of the width when it goes off screen).
- The 2D/3D simulation follows the red line: moving the red line with the mouse shows the lighting at that moment.

- When paused, the scene's final state is displayed in full (the incoming fade is considered finished), which lets you make adjustments and move around in the timeline.

## The tracks

A track is a horizontal line of the timeline. Each track can take any type of block: there is no track reserved for light or for sound. The track's name is shown on the left.

Tracks are there to organise things, but they also have real effects:

- two light blocks that overlap **on the same track** make a mix (crossfade from one to the other); on two different tracks, they are layered (see Fades and mixes);
- two sounds that overlap on the same track crossfade;
- the **Stop sound** block can cut only the sounds started from certain tracks, designated by their name;
- a **Blackout** block only turns off the fixtures lit by the blocks of its own track (or everything, if its track has no light).

### Adding a track

- Click **+ track**.
- Or, if the scene has no track yet, drop an item on the grey strip “Drag an item here to create a track”: DreamShow creates “Track 1” and places the block on it. If you cancel the block's settings window, the track is not created.
- The **Add to the timeline** button of the **Media** list creates a “Media” track if needed (or “DMX” for a DMX recording).

### The current track

A click on a track's name makes it the **current track**: its name turns orange, preceded by “ ▸ ”. Tooltip: “Click = current track (receives new blocks) · right-click = rename / delete”.

The current track receives the blocks added by the **Add to the timeline** button (**Media** list). Blocks dragged with the mouse, on the other hand, go onto the track where you drop them.

### A track's menu (right-click on its name)

- **Move the track up** / **Move the track down**: changes the order of the tracks (greyed out for the first or the last). The order matters: when two tracks set the same fixture at the same moment, the lower track wins (see Light and effect blocks).
- **Duplicate the track**: copies the track and all its blocks just below it; the copy gets a unique name (“Light (2)”…).
- **Copy the track**: puts the track and its blocks in memory.

- **Paste the track**: adds the last copied track at the bottom of the displayed scene (greyed out as long as nothing has been copied). It works from one scene to another: this is how you reuse a track in another scene. The pasted track becomes the current track.
- **Rename the track…**: asks for the new name (“Name:”). The scene's **Stop sound** blocks that targeted the old name are updated.
- **Delete the track**: deletes the track and its blocks. If the track contains blocks, DreamShow asks for confirmation (“Delete track “…” and its N item(s)?”).

### Two-level tracks

When a track contains both **Stop sound** or **Sound level** blocks and other blocks, it is displayed on two levels: ordinary blocks at the top, **Stop sound** and **Sound level** blocks just below. That way a level setting never hides the sound it goes with.

## Blocks

### Dropping a block

All blocks come from the **Libraries** panel (on the right): you **drag** them with the mouse from a list and **drop** them on a track, at the moment you want.

- While dragging, a dotted rectangle shows where the block will land.
- The drop point is rounded to the tenth of a second and **snaps**: if it falls within 8 pixels of the start or end of another block of the scene (on any track), or of 0 s, it sticks to it. A dotted yellow line shows the snapping.

What each list drops:

| List | Block created | Starting duration |
|---|---|---|
| **Groups** | light block on this group, with the group's setting in the final state | 10 s |
| **Effects** | library effect block | 10 s |
| **Trajectories** | trajectory block | the trajectory's duration (10 s if unknown) |
| **Media** (sound or video) | media block | the real duration of the file (30 s if it is unknown) |
| **Media** (.DmxStreaming, .bin, .dmx) | DMX recording block | the real duration of the recording (10 s otherwise) |
| **Others** | depends on the item (see below) | depends on the item |

Some items of the **Others** list open their settings window straight away (**Go to scene…**, the two parallel sequences, the two waits, **Backstage display…**, **Robot…**, **Mask…**, **Dynamic effect…**, **Pixel mapping…**). If you close this window without confirming, no block is created.

### Moving a block

- **Drag the block** horizontally: its start changes in steps of 0.1 s. During the gesture, the block becomes transparent with a dotted outline.
- The move **snaps**: the start **or the end** of the block sticks to the start or end of another block of the scene (all tracks) or to 0 s.
- **To change track**, drag the block vertically over another track: a dotted rectangle appears on that track; on release, the block is moved there.

### Resizing a block

Blocks that have a duration (light, effect, trajectory, media, DMX recording, dynamic effect, pixel mapping, **Blackout**, **Silence**) have a light handle on their right edge (the pointer becomes a double arrow). Drag it to change the duration, in steps of 0.1 s (minimum 0.1 s). During the gesture, the block shows its duration ("↔ 12.5 s").

The other items of the **Others** list are **instant events**: only their start matters, their width is only there so you can read their text. They have no handle.

### Double-click

- On an **effect** block: switches to effect editing mode (orange "EFFECT MODE" banner), see Light and effect blocks.
- On a **trajectory** block: opens the trajectory editor, see Trajectories.
- On a **dynamic effect** block: opens the **Dynamic effect** window, see Dynamic effects.
- On a **pixel mapping** block: opens the **Pixel mapping** window, see Pixel mapping.
- On a **Robot** or **Mask** block: opens its settings window.
- On all the others: opens the **Block properties** window (or the event's own window: **Go to scene**, **Wait for a key**…).

### A block's menu (right-click)

- **Properties…**: the block's settings window (start, duration, fades, and the settings specific to its type).
- **Edit the effect…** (effect block only): switches to effect editing mode.
- **Edit the trajectory…** (trajectory block only): opens the trajectory editor.
- **Replace with the current selection + settings** (light block only): see Light and effect blocks.
- **Delete**: removes the block from the timeline, without confirmation.

### The Block properties window

For a light, effect, trajectory, dynamic effect or pixel mapping block, a **Blackout** or a **Silence**:

- **Start (s)**: start time, in seconds from the start of the timeline.
- **Length (s)**: length of the block.
- **Fade in (s)** and **Fade out (s)**: gradual rise and fall of the block (see Fades and mixes). If the two fades add up to more than the duration, DreamShow shortens the fade out (and the fade in if on its own it exceeds the duration).

Media blocks have extra settings, described in Sound and video on the timeline. A DMX recording block only has **Start (s)** and **Length (s)**.

Negative or unreadable values are ignored (the old value is kept). Both full stop and comma are accepted as the decimal separator.

### What a block shows

- Its **text**: the name of the group or effect, of the media file, or a summary of the event ("→ 3 · Final", " ⏳ wait 18:30", "♪ Level 80 %"…).
- Its **fades**: a dark triangle in the left corner (fade in) and in the right corner (fade out), to scale.
- A **cross** (as in video editing software) where two blocks overlap on the same track. Between two light blocks, it signals a mix; between two sounds, a crossfade; between two **Sound level** blocks, a level transition.
- Its **tooltip** (on hover): type, start, duration, fades, playback position in the file.
- A **red block preceded by** ⚠ signals a problem: media missing from the show's media folder ("MEDIA NOT FOUND in the show's media folder"), effect deleted from the library ("EFFECT NOT FOUND in the library"), missing trajectory ("TRAJECTORY NOT FOUND"), jump or parallel sequence whose target scene no longer exists.

The block colours are described in Light and effect blocks.

## Block types

| Block | Where it comes from | Page |
|---|---|---|
| Light (group) | **Groups** list | Light and effect blocks |
| Effect | **Effects** list | Light and effect blocks |
| Trajectory (↝) | **Trajectories** list | Trajectories |
| Sound, video | **Media** list | Sound and video on the timeline |
| DMX recording (▦) | **Media** list | Sound and video on the timeline, DMX stream by an AI |

| | | |
|---|---|---|
| Blackout (⚫) | **Others** list | Light and effect blocks |
| Stop sound, Sound level, Silence | **Others** list | Sound and video on the timeline |
| Wait for a key…, Wait for a time, a date, the sun… | **Others** list | Waits |
| Go to scene… | **Others** list | Jumps and scene labels |
| Start / End a parallel sequence… | **Others** list | Parallel sequences |
| Dynamic effect… (∿) | **Others** list | Dynamic effects |
| Pixel mapping… (⣿) | **Others** list | Pixel mapping |
| Backstage display… (▭) | **Others** list | see below |
| Robot… (🤖), Mask… (🎭) | **Others** list | see below |

### Backstage display…

Sends a message to the information screens in the wings and dressing rooms (backstage displays) at the moment the red line reaches it. Its window (**"Afficheur backstage"** (Backstage display)) contains the text of the message (**Message (empty = clears the screen):**), buttons to insert automatic values (**{heure}**, **{durée}**, **{fin}**, **{scène}**, **{suivante}**: time, length, end, scene, next) and colours (**{rouge}**, **{vert}**, **{jaune}**, **{orange}**, **{bleu}**, **{blanc}**: red, green, yellow, orange, blue, white; **{alerte}** for a red background), then the **Target displays:** list to tick. A block with no target does nothing. Displays are declared in **Venue > Backstage displays…**. Details: Backstage displays.

### Robot… and Mask…

A **Robot…** block gives an order to a humanoid robot of the venue (gesture, phrase, dance…), a **Mask…** block controls an LED mask (face, mouth animated on a sound…). Their window (**Robot block** or **Mask block**) asks for the target device, the action and its setting. A robot only moves if the operator has armed it. Robots and masks are declared in the **Venue** menu. Details: Humanoid robots and LED masks.

## Disengage auto-fade

The **Disengage auto-fade** box (on the right of the toolbar) cuts automatic follow-ons. Its tooltip: "Ticked: no scene chains on its own (auto-fade / chained), and a GO arrives on the scene WITHOUT playing its timeline (final state frozen, no sound, no blocks). Everything is advanced and played by hand — useful to edit and test."

What it does exactly when it is ticked:

- Scenes set to **Auto** or **Chained** no longer move on to the next one by themselves, nor does the **max length** of a **Loop** scene (the loop itself keeps going round).
- Parallel sequences no longer chain from one scene to another either.
- In the editor, a GO (**GO (Space)** button or space bar) arrives on the next scene **paused, at 0 s**: you see its final state, no sound plays; click ▶ to play its timeline.
- In Show mode, a GO still starts the next scene normally: there, the box only blocks automatic follow-ons.

The same box exists in the cue window; the two are linked. It is unticked every time DreamShow starts.

## What happens at the end of the timeline (in the editor)

When you play a scene with ▶:

- **Scene on "Manual GO", without Loop**: playback stops by itself just after the last block, like a media player at the end of a track. The red line stays parked at the end, ▶ goes back to the play position. To play again, click ⏮ (or in the ruler), then ▶. An empty scene stays at 0 s.
- **Scene on Auto or Chained**: playback continues and the scene moves on to the next one by itself, with its fade (unless **Disengage auto-fade** is ticked). The selection in the **Scenes** panel follows (if the **LIVE** button is on).
- **Scene on Loop**: the timeline starts again from the beginning, until GO or until the max length.

Stopping at the end of the timeline pauses **the whole engine**, parallel sequences included. To watch a parallel sequence run for a long time, switch to Show mode (see Parallel sequences).

The **GO (Space)** button stays greyed out as long as the timeline of the current scene has not finished (except for a **Loop** scene, which GO can always leave, and except before the first scene). The same rule applies in the cue window.

## Playback in the editor and in Show mode

| | **In the editor** | **In Show mode (cue window)** |
|---|---|---|
| Choosing a scene | the engine jumps to it (**LIVE** button on), without a fade, at 0 s | you move on with GO; ◀ Previous goes back |
| Playback | paused by default; ▶ plays | always playing |
| End of timeline | stops at the end of the timeline (scene on Manual GO) | the red line stays at the end of the last block; the scene holds its final state until GO |
| GO | plays the next scene (fade then timeline), or shows it paused if **Disengage auto-** | always plays the next scene |

| | | |
|---|---|---|
| | **fade** is ticked | |
| DMX output | follows the **LIVE** button | always active |

When the cue window opens, the show starts again from the “Ready” state (black scene), or resumes where you left it if you closed the cue window in the middle of the show. When it closes, the editor goes back to the selected scene, paused. See Show mode and the cue window.

## Step by step

### Building a first timeline

1. In the **Scenes** panel, select the scene.
2. In **Scene settings**, set the lighting of a group (for example “Front” at 80 % in amber): this is its final state.
3. Drag the “Front” group from the **Groups** list onto the strip “Drag an item here to create a track”, around 2 s. DreamShow creates “Track 1” and a 10-second block that takes the group's setting.
4. Right-click the block > **Properties…**: set **Fade in (s)** to 2 and **Fade out (s)** to 3, then **OK**.
5. Drag a sound from the **Media** list onto the same track or onto a new one (**+ track**), at 0 s.
6. Click ⏮ then ▶: the red line moves, the sound starts, the light rises and falls in the simulation. Playback stops after the last block.

### Reorganising a timeline

1. Rename the tracks (right-click on the name > **Rename the track…**): “Front”, “Backlights”, “Music”.
2. Drag a block to another track to move it.
3. Butt the blocks end to end with the help of snapping (yellow line).
4. Zoom with Ctrl + wheel to place a short block precisely.

### Reusing a track in another scene

1. Right-click the track's name > **Copy the track**.
2. Select the other scene in the **Scenes** panel.
3. Right-click the name of any track > **Paste the track** (if the scene has no track, first add one with **+ track**).

## Good to know

- There is no undo (Ctrl+Z) in the timeline. **Delete** removes a block without confirmation. Save often (Ctrl+S).
- The Delete key does not delete a block: use right-click > **Delete**.

- **+ track** numbers tracks according to the number of tracks: after a deletion, two tracks can end up with the same name. Rename them, especially if you use **Stop sound** per track.
- With the **LIVE** button off, choosing a scene in the **Scenes** panel no longer makes the engine jump to it; ⏮ or a click in the ruler still do.
- A trajectory block is stretched over the length of the block: a longer block plays the trajectory more slowly. The last movement of a track keeps its final position until the end of the scene. See Trajectories.
- The panel can be detached into a floating window (⧉ button in its header), see Main screen overview.

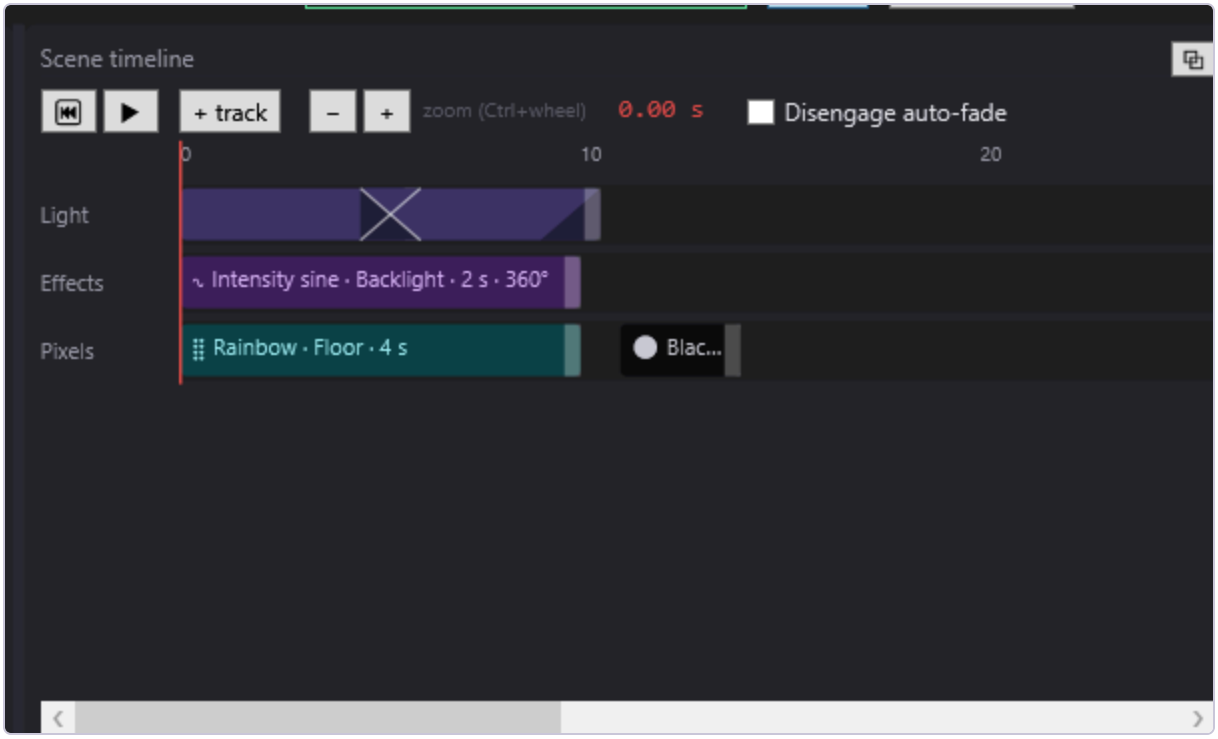
## See also

- Light and effect blocks
- Fades and mixes
- Sound and video on the timeline
- Waits (key, time, date, sun)
- Jumps and scene labels
- Parallel sequences
- Scene settings
- The Scenes panel
- Libraries
- Show mode and the cue window

# Light and effect blocks

Light blocks make the lighting change during the scene: a group that rises then falls, a colour that passes through, a profile spot that comes on for the length of a line. They are placed on the timeline, over the scene's **final state**, and last as long as you give them. This page also describes the **Blackout** block and the block colours.

**To open it:** drag a group from the **Groups** list, or an effect from the **Effects** list (**Libraries** panel), onto a track of the **Scene timeline** panel. The **Blackout** block is in the **Others** list.




## The two kinds of light block

### The group block

This is a block that contains its own setting, for one or more groups or fixtures.

- **Creation**: drag a group from the **Groups** list onto a track. The block lasts 10 s, with no fade, and bears the group's name.
- **Its initial content**: the setting of this group in the final state of the displayed scene (intensity, colour, position, special channels such as the colour wheel or the zoom). If the group is not in this scene's final state, DreamShow takes its setting from another scene of the show; if it finds none, the block sets the group to 100 % white (and opens the moving heads' shutters).

- **The block keeps a copy** of these values: changing the final state afterwards does not change the block.
- **Palettes follow**: if the group used colour, position or beam palettes, the block keeps them. Changing a palette therefore updates the block too, and with a position palette each moving head keeps its own position.

### The effect block

An **effect** is a named setting (one or more groups with their settings) stored in the **Effects** list and reusable in every show.

- **Creating an effect**: tick the groups or fixtures in **Selection — groups and fixtures**, set them, then click **+ effect (selection)** in the **Effects** list; give a name (**New effect** window, “Effect name:”). Two effects cannot have the same name (“An effect already has that name.”).
- **Creating the block**: drag the effect from the **Effects** list onto a track. The block lasts 10 s and bears the effect's name.
- **The block follows the effect**: it contains no copy, it refers to the effect by its name. Changing the effect changes every block that uses it, in every show.
- If the effect is deleted from the library, the block turns red with ⚠ (“EFFECT NOT FOUND in the library”) and no longer lights anything.

## Changing a block's content

### Effect block: effect mode

Double-click the block (or right-click > **Edit the effect…**). An orange banner appears at the top of **Scene settings**: “EFFECT MODE — “name”: the settings change the effect, not the sequence”. The effect's groups are ticked, and the right-hand list becomes “Contents of effect “name””.

1. Set the sliders (Dim, R, G, B, Pan, Tilt, specific settings), as for a final state. The simulation shows the effect over the scene.
2. Click ✓ **Finish editing the effect**: the effect is saved and you go back to sequence mode.

See Selecting fixtures and adjusting the light for the sliders and Libraries for the **Effects** list (double-click = edit, right-click > **Delete the effect**).

### Group block: Replace with the current selection + settings

A group block does not open for editing. To change its content, right-click > **Replace with the current selection + settings**: for each ticked group or fixture, the block receives its setting from the final state, with the intensity, colour (R, G, B) and position (Pan, Tilt, for moving heads) of the current sliders. The block's text becomes the list of targets (“Front + Backlights”). With nothing ticked, DreamShow replies “Tick some targets and set the panel first.”.

Caution: in sequence mode, the sliders change the scene's **final state**. If the block must differ from the final state, remember to put the final state back as it was after the replacement, or use an effect instead (see the step by step).

## A block's settings (Properties…)

Right-click > **Properties…** (or double-click on a group block) opens **Block properties**:

- **Start (s)**: the moment the block starts.
- **Length (s)**: length of the block. It can also be set with the mouse using the handle on the right edge.
- **Fade in (s)**: rise time. 0 = the block takes over at once.
- **Fade out (s)**: fall time, which ends at the end of the block. 0 = the block disappears at once.

The fades are drawn as dark triangles in the corners of the block. The two fades together cannot exceed the duration: DreamShow corrects this if needed.

## How a block and the final state combine

The frame sent to the fixtures is built at every moment (40 times a second), in layers:

1. **The scene's final state**, reached through the scene's incoming fade.
2. **Light and effect blocks**, track after track from top to bottom, and on each track in order of start time. Each block is laid over what is below it.
3. **Trajectories, dynamic effects and pixel mappings**, over all the light blocks.
4. **Blackout blocks**.
5. **DMX recordings** (▦ blocks).
6. **Parallel sequences**, mixed with the cue list (see Parallel sequences).
7. The general blackout and the manual controls override everything.

For each channel a block sets:

- **Intensity channels** (dimmer, red, green, blue, white colour mixing…): the value moves gradually from what is below to the block's value during the fade in, stays at the block's value, then returns to what is below during the fade out.
- **Other channels** (pan, tilt, zoom, colour and gobo wheels, shutter…): they take the block's value **from the very first moment** of the block, without waiting for the end of the fade, and keep it until the end of the block. A moving head therefore gets into position before the light comes up.
- Channels the block does not set stay as they are below.

At the end of the block, what is below takes over again:

- intensity returns to the final state (after the fade out);

- a position set by the final state returns at once to the final state's value; a position the final state does not set stays where the block left it (a moving head does not swing back to its end stop);
- a fixture absent from the final state drops back to zero (“not listed = to zero”, except devices set to keep their value, such as a winch).

Examples:

- Final state: Front at 30 %. Block: Front at 100 %, fade in 2 s, fade out 3 s. → Front rises from 30 to 100 % in 2 s, stays there, then falls back to 30 % in 3 s.
- Final state: Front at 80 %. Block: Front at 0 %, 1 s fades. → a dip: Front goes down to 0 then back up to 80 %.
- Empty final state. Block: blue Backlights at 100 %. → the backlights come on during the block, then go out.

### When two blocks set the same fixture

- **On the same track and overlapping**: this is a **mix**, one blends into the other (colour and position included). See Fades and mixes.
- **On two different tracks**: they are layered; at full intensity, **the block on the lowest track wins**. During its fade in, it blends with the one above. Use **Move the track up** / **Move the track down** to choose which comes in front.

## The Blackout block

The **Blackout** block (text “⚫ Blackout”, black block) turns the light off for its whole duration, without touching positions, colours, wheels or shutters: moving heads keep their aim, only the intensity drops to 0.

- **Creation**: drag **Blackout** from the **Others** list onto a track. It lasts 10 s and can be resized like a light block.
- **What it turns off**:
  - placed on a track that also carries light blocks or trajectories, it turns off **only the fixtures that this track lights**;
  - placed on a track with no light, it turns off **all the fixtures** (general blackout for its duration).
- It also turns off the final state of these fixtures, and whatever trajectories and dynamic effects are doing.
- **Its fades**: you can set **Fade in (s)** and **Fade out (s)** in **Properties...**. In addition, on its own track, DreamShow adds automatic fades:
  - if a light on the track is on when the Blackout starts, the light goes down to black during the overlap, instead of cutting;

    - if a light on the track continues after the end of the Blackout, it comes back up during the overlap.
- Parallel sequences, mixed in afterwards, are not turned off by a Blackout in the cue list.

## Block colours

| Block colour | Type |
|---|---|
| Purple | light block (group) |
| Green | effect block |
| Orange-brown (↝) | trajectory |
| Deep purple (∿) | dynamic effect |
| Blue-green (⣿) | pixel mapping |
| Blue | sound or video |
| Plum (▦) | DMX recording |
| Black ( ⚫ ) | Blackout |
| Burgundy | Stop sound, Sound level, Silence |
| Dark blue-green (→) | Go to scene |
| Blue (⇉, ⇥) | parallel sequences |
| Dark green (⌨) | Wait for a key |
| Blue-grey ( ⏳ ) | Wait for a time, a date, the sun |
| Blue-violet (▢) | Backstage display |
| Brick red ( 🤖 ) | Robot |
| Mauve ( 🎭 ) | Mask |
| Bright red with ⚠ | block in error (file, effect or scene not found) |

In the **Libraries** panel, a small coloured square in front of **Groups**, **Media**, **Effects**, **Trajectories** and **Others** recalls the family of blocks of each list.

## Step by step

### Bringing up a profile spot for the length of a line

1. In the scene's final state, leave the “Special” profile off (or remove it with **Remove from the final state**).

2. Create a “Warm special” effect: tick “Special”, set Dim 90 %, a warm colour, then **+ effect (selection)**, name “Warm special”.
3. Drag “Warm special” from the **Effects** list onto a track, at 12 s.
4. Right-click > **Properties…**: **Length (s)** 20, **Fade in (s)** 3, **Fade out (s)** 4. **OK**.
5. ⏮ then ▶: the special rises at 12 s, holds, and goes out over 4 s at 32 s.

### Adding a splash of colour on a background that is already lit

1. Final state: “Cyc” in blue at 60 %.
2. Drag “Cyc” from the **Groups** list onto a track: the block takes the blue at 60 %.
3. Tick “Cyc”, set the sliders to a red at 100 %, then right-click the block > **Replace with the current selection + settings**.
4. Set the sliders of “Cyc” back to blue at 60 % (this is the final state you have just changed).
5. Set 1 s fades: the cyc turns red during the block, then returns to blue.

### Making a blackout in the middle of a scene

1. Drag **Blackout** from the **Others** list onto a track **with no** light block, at the moment you want.
2. Set its duration to the length of the blackout.
3. In **Properties…**, give a **Fade in (s)** (going down) and a **Fade out (s)** (light coming back).

## Good to know

- A Blackout placed **on the same track** as a longer light block, and entirely inside it, gets automatic fade in and fade out as long as itself: the light then only goes down halfway in the middle of the Blackout. For a clean blackout, put the Blackout on another track and set its fades yourself.
- Blocks have no “keep the setting after the end” button. For a setting to stay, put it in the scene's final state. In a show imported from ppshow, however, the last block of a series of steps keeps its setting after it ends.
- For a moving head to give light, its shutter must be open: blocks created from a group of moving heads open it by default.
- The light blocks of a track only mix with each other: a sound or a video placed between two light blocks does not get in the way of the mix.

## See also

- The scene timeline
- Fades and mixes
- Selecting fixtures and adjusting the light
- Scene settings
- Libraries

- Palettes
- Dynamic effects
- Pixel mapping
- Trajectories

# Fades and mixes

DreamShow moves the light from one state to another in three ways, which do not behave in the same manner: the **scene fade** (when going from one scene to the next), the **block fade** (its rise and fall) and the **mix** (two blocks that overlap on the same track). Telling them apart avoids unpleasant surprises: a moving head that jumps instead of gliding, or a colour change that makes a dip.

**To open it:** the scene fade is set in **Scene settings**, field **Fade to the next (s):**. A block's fades are set in its **Properties…** window (right-click on the block). The mix is made by overlapping two blocks on the same track of the **Scene timeline** panel.

## In summary

| | **Scene fade** | **Block fade** | **Mix** |
|---|---|---|---|
| Where it is set | **Fade to the next (s):** of the scene being left | **Fade in (s)** / **Fade out (s)** of the block | overlap of two light blocks on the same track |
| When | on GO, or on automatic chaining | at the start and end of the block | during the overlap |
| Intensity and colour mixing | gradual | gradual | gradual |
| Position, zoom, focus (moving heads) | gradual: the heads glide | immediate: block's value from its start | gradual: the heads glide |
| Wheels, gobos, shutter | change at the start of the fade (or at the end if they go to 0) | immediate | gradual (careful, see below) |

## Channel families

To know what “fades” and what “jumps”, DreamShow sorts each channel of each fixture into a family, from its definition in the fixture library:

- **Intensity**: dimmer, colour mixing (red, green, blue, white, amber, UV, cyan, magenta, yellow…), cells of a bar or matrix.

- **Continuous parameters**: pan, tilt, zoom, focus, iris, frost, and a simple colour channel (a level from 0 to 255 with no named positions).
- **Discrete parameters**: shutter and strobe, colour and gobo wheels, prisms, macros, and all the other channels.

16-bit channels (pan and fine pan, for example) are always handled together, as a single value: a moving head does not shake during a fade.

## The scene fade: “Fade to the next”

The **Fade to the next (s):** field belongs to the scene you are **leaving**. Its tooltip: “As in ppshow: length of the fade applied when LEAVING this scene (on GO or automatically), towards the state of the next scene. The timeline of the next scene starts when the fade ends.”

What happens on GO (or on automatic chaining):

1. DreamShow starts from what is **really on stage** at that instant (final state of the scene being left, plus the blocks still active).
2. For the length of the fade, everything glides towards the **final state of the next scene**:
    - intensity and continuous parameters pass gradually from one value to the other (the moving heads move during the fade);
    - discrete parameters take their new value **at the start** of the fade; if they go to 0 (shutter closed, wheel at rest), they only change **at the end**, so that the light does not disappear before its intensity has come down.
3. At the end of the fade, the next scene's timeline starts at 0 s.

During the fade, the editor and the cue window show “fade 2.4 s” (remaining time). The cue window also recalls a scene's fade (“→ fade 5 s”).

What does **not** fade: choosing a scene in the editor's **Scenes** panel, **◀ Previous** in the cue window, a **Go to scene…** block, resuming after a crash. These jumps are instantaneous.

The videos of the scene being left stop at the start of the fade; the sounds carry on (see Sound and video on the timeline).

## The block fade

**Fade in (s)** and **Fade out (s)** set the rise and fall of a light, effect, trajectory, dynamic effect or pixel mapping block. They concern **intensity only**:

- the block's intensity rises from what is underneath during the fade in, and falls back to it during the fade out;
- position, zoom, wheels and shutter take the block's value **from its very first instant**, and keep it until its end.

This is deliberate: a moving head gets into place while its light rises, instead of sweeping across the stage; and a shutter never runs through its strobe values.

The fade out ends at the end of the block. The sum of the two fades never exceeds the block's length.

## The mix

When **two light or effect blocks overlap on the same track**, the first blends into the second, like two clips that overlap in a video editing program. The timeline draws a cross on the overlap.

During the overlap:

- **What both blocks set** (the same channels of the same fixtures) glides from the first block's value to the second's, **whatever the channel's family**: intensity, colour, position, zoom… The first block stays at full intensity during this time (no dip in the middle).
- **What only the first block sets** fades out during the overlap: its fade out is replaced by the length of the overlap.
- **What only the second block sets** rises during the overlap: its fade in is at least the length of the overlap.

The mix only happens between two **consecutive blocks of the same track**. On two different tracks, two blocks do not blend: they are layered (see Light and effect blocks).

Other overlaps also make an automatic fade, each within its own family:

- two **sounds** on the same track: crossfade (see Sound and video on the timeline);
- two **Sound level** blocks on the same track: the level passes from one to the other;
- a **Blackout** and a light on the same track: the light goes down to black or comes back up from it.

## Examples

### A colour that slowly turns

“Cyclo” track: block A, red cyclorama at 100 %, from 0 to 10 s; block B, blue cyclorama at 100 %, from 8 to 20 s. → From 8 to 10 s, the cyclorama goes from red to blue through purple, without dipping. The cross covers 8–10 s.

If B is on another track (lower down), there is no mix: at 8 s, B is laid on top of A. With no fade in, the cyclorama goes blue at once; with a fade in, the colours' intensity glides towards blue during that fade.

### Moving heads that glide from a singer to the piano

“Moving heads” track: block A, heads aiming at the singer, from 0 to 15 s; block B, heads aiming at the piano, from 13 to 30 s. → From 13 to 15 s, the heads move from the singer to the piano, with the beam on.

The same without overlap (A from 0 to 15 s, B from 15 to 30 s with a 2 s fade in): at 15 s, the heads jump straight to the piano, and only their intensity comes back up over 2 s.

### A day-to-evening change on GO

Scene 1 “Day”: **Fade to the next (s):** = 8. Scene 2 “Evening”: warmer and lower final state, moving heads repositioned. → On GO, over 8 s, the light goes down and warms up, the heads glide to their new positions. A gobo change, on the other hand, happens at the start of the fade. The “Evening” timeline then starts.

### A deliberate “blackout in between”

“Front” track: block A from 0 to 10 s with **Fade out (s)** 2; block B from 10 to 20 s with **Fade in (s)** 2, with no overlap. → The front light goes down to the final state (or to black if the final state does not light it) between 8 and 10 s, then comes back up with B's setting between 10 and 12 s.

## Step by step: setting a crossfade between two blocks

1. Place the two blocks on the **same track**.
2. Drag the second block to the left until it overlaps the end of the first by the length you want (for example 3 s). A cross appears.
3. Leave the second block's **Fade in (s)** at 0 (the overlap does the work).
4. ⏮ then ▶ to check in the simulation.

## Good to know

- During a mix, even the colour and gobo wheels, and the shutter, glide from one value to the other if the two blocks set them differently: the wheel scrolls, the shutter may run through strobe values. Give both blocks the same values for these channels, or separate the blocks.
- If the second block of a mix has a fade in longer than the overlap, the fixtures set by both blocks stay at the level reached at the end of the mix; those that only the second block lights follow its fade in to the end.
- In a parallel sequence, two overlapping blocks only make an intensity fade: colour and position change at once.
- During a pause (editor stopped, wait block), the scene fade is considered finished: the editor shows the final state directly.

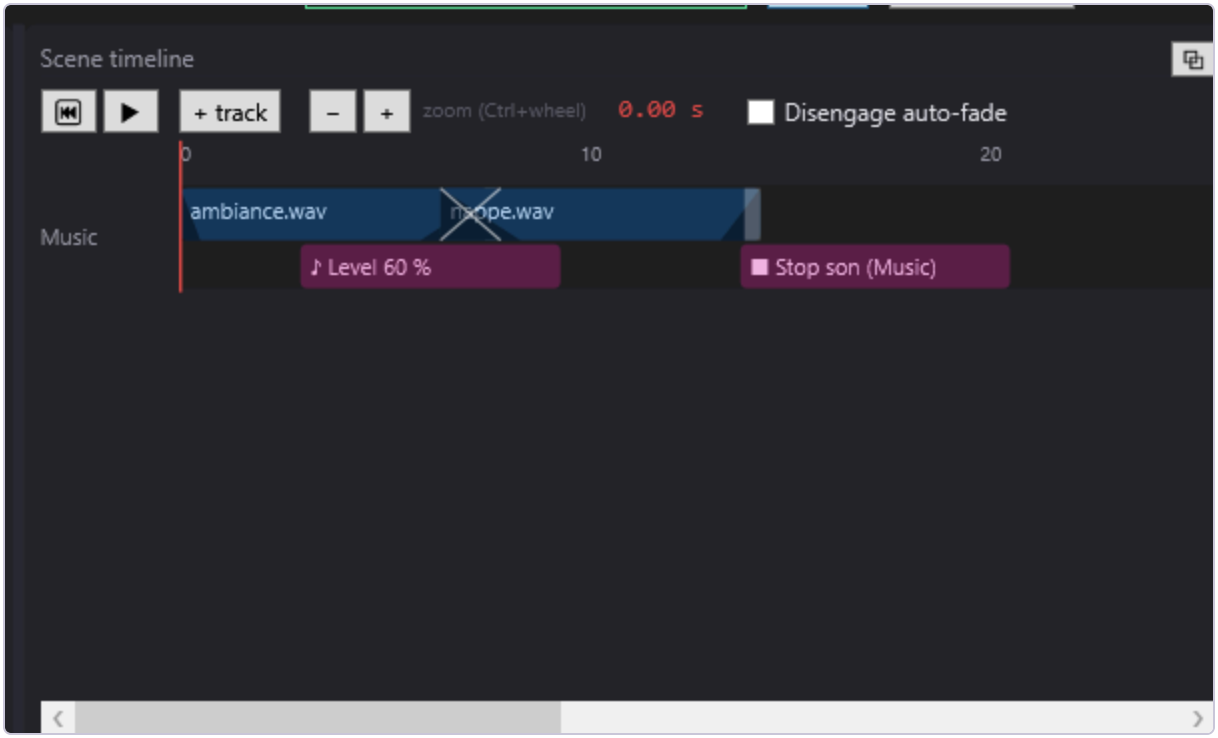
## See also

- Light and effect blocks
- The scene timeline
- Scene settings
- Sound and video on the timeline
- Trajectories
- Glossary

# Sound and video on the timeline

The sounds and videos of your show are placed on the scenes' timelines, like light blocks: they start when the red line reaches them, with their fades, and follow the red line when you move around the timeline. This page also describes the **Stop sound**, **Silence** and **Sound level** blocks of the **Others** list, and DMX recordings.

**To open it:** drag a file from the **Media** list (**Libraries** panel) onto a track of the **Scene timeline** panel, or select it and click **Add to the timeline**. **Stop sound**, **Silence** and **Sound level** are dragged from the **Others** list.




## Media files

A show's media are stored in its media folder (**File > Open the show's media folder**). The **Media** list shows this folder.

- **Import media...**: copies one or several files into the show's media folder.
- ↻: “Refresh the media list”.
- **Add to the timeline**: drops the selected medium at 0 s on the current track (or on a “Media” track, created if needed). If nothing is selected, a file-choice window opens; the chosen file is copied into the show's media folder, then placed on the timeline.

- Right-click on a medium: ▶ **Play** (listens to or displays the file as a test), ■ **Stop**, **Edit with GoldWave**, **Open the media folder**, **Delete the media**. See Libraries.

DreamShow recognises the type from the extension:

- **Sound**: .mp3, .wav, .flac, .ogg, .m4a, .aac, .wma. Played by DreamShow itself, on the Windows sound output.
- **DMX recording**: .DmxStreaming, .bin, .dmx (see below).
- **Video**: all the other files (.mp4, .avi, .mov, .mkv…). Played by DreamShow's video player (ppvideoplayer) on a screen of the venue, or by an IP video player.

## The media block

A media block is blue and bears the file name. Its initial length is the real length of the file (30 s if unknown; it corrects itself as soon as the file has been read once). If the file is missing from the media folder, the block is red with ⚠ (“MEDIA NOT FOUND in the show's media folder”).

### The settings (Properties…)

Right-click > **Properties…** or double-click:

- **Start (s)**: the moment the medium starts.
- **Length (s)**: playing time. At the end of the block, the medium stops, even if the file is longer. A block longer than the file does not extend the file.
- **Fade in (s)**: the sound rising from silence, or the video appearing from black.
- **Fade out (s)**: the sound dropping to silence, or the video fading to black; it ends at the end of the block. For a video, 0 lets the video player apply its default setting.
- **Play from (s)**: starting position in the file (for example 12 to skip the first 12 seconds). 0 = from the beginning.
- **“Chargement”** (Loading): when the file is opened in advance, so that the start is immediate:
  - **When the scene loads** (default): opened when the scene becomes the current scene, or the one before it; released when it is left;
  - **When the project loads**: opened when the show is opened and kept;
  - **Without preloading**: opened only at the moment it is played (a slight delay is possible).
- **Screen** (videos only): the screen of the venue where the video is played. **(file tag / default)** lets the video player choose from the file name, otherwise its default screen; the other choices are the screens declared in **Venue > Screens…**. See Video screens.

### Hints in a video file's name

The video player reads hints written in brackets in the file name, for example `generique (M=2) (FI=1500) (LOOP).mp4`. They apply when the block leaves **(file tag / default)**.

| Hint | Effect |
|---|---|
| `(M=2)` | plays the video on Windows monitor 2 |
| `(LOOP)` | plays the video in a loop |
| `(FI=1500)` | fade in of 1,500 milliseconds (1,000 by default) |
| `(FO=800)` | fade out of 800 milliseconds (500 by default) |
| `(PI=500)` | when paused, the video gradually slows down for 500 milliseconds before stopping |
| `(PO=500)` | when resumed, the video gradually speeds up for 500 milliseconds |
| `(AMBI)` | enables ambilight: see `AMBILIGHT=` in Scene settings |

The fade in and fade out set on a timeline block replace those in the file name.

There is no volume or loop setting per block:

- the volume is global (sliders of the **Audio** panel and **Sound level** blocks);
- to repeat a sound or a video, tick **Loop** in the scene settings: on each pass, the timeline restarts from 0 and the media restart (see Jumps and scene labels).

### Sounds

- **They follow the red line**: clicking in the ruler in the middle of a sound makes it play from that point; pausing pauses it.
- **Two sounds that overlap on the same track** make a crossfade: the first goes down and the second comes up during the overlap (a longer fade that you have set is kept). A cross appears on the overlap.
- **On two different tracks**, two sounds play together. The same file can even play twice at the same time from two tracks.
- **A sound carries on after a scene change**: a GO does not cut it. If it is still playing when you move on to the next scene, it continues to the end of the file (if it has a fade out, it fades out where its block ended). To stop it, drop a **Stop sound** or a **Silence** in the next scene.
- The **Mute the sounds** button of the **Audio** panel (and of the cue window) stops all sounds immediately. See Log, Audio and Clock.

### Videos

- A video stops when its scene is left (the player fades to black).
- On a Windows screen, the video follows the red line: clicking in the ruler in the middle of the block restarts it at the right frame; when paused, the picture stays frozen at the position of the red line.
- On an IP video player, the video starts from the file attached to a player slot: the block then shows "· slot A" next to the name. The file must be associated with a slot in **Venue >**

**Screens...**. An IP player does not follow the red line's movements.
- Two videos that overlap do not make an automatic crossfade: each keeps its own fades.
- With a screen placed in the venue, the video also appears in the 2D/3D simulation, locked to the red line. See 2D and 3D simulation.

## Stop sound

Cuts sounds at a precise moment.

- **Creation**: drag **Stop sound** from the **Others** list onto a track. It is an instantaneous event.
- **Settings** (**Stop sound — tracks to stop** window):
  - **Start (s)**: the moment of the cut.
  - **Length (s)**: the block's width on screen only; the cut is immediate.
  - **All tracks** (ticked by default): cuts all sounds.
  - one box per track of the scene (active when **All tracks** is unticked): cuts only the sounds started from the ticked tracks. If none is ticked, all sounds are cut.
- The block shows "■ Stop sound (all)" or the list of targeted tracks.
- Tracks are recognised by their **name**: a sound started in a previous scene from a track of the same name is cut too. This is how to stop, in scene 3, the music started by the "Music" track of scene 2.
- A **Stop sound** and a new sound placed at the same moment: the old sound is cut first, then the new one starts.

## Silence

Silences all sounds, with a fade if a sound of the scene is playing.

- **Creation**: drag **Silence** from the **Others** list onto a track. The block lasts 3 s and can be resized.
- **What it does**: at its start, it stops **all** sounds. If a sound of the scene is playing at that moment, all sounds fade down over the length of the block (or until the end of that sound if it ends sooner). Otherwise, the cut is immediate.
- The block shows " Silence — fade 3 s".
- Its **Properties...** window offers **Fade in (s)** and **Fade out (s)**, but only the **Length (s)** counts.

## Sound level

Changes the overall volume of DreamShow's sounds, as if you moved the L and R sliders of the **Audio** panel (they do move on screen, too).

- **Creation**: drag **Sound level** from the **Others** list onto a track. The block takes the current levels of the **Audio** panel and lasts 5 s.
- **Settings** (**Sound level** window):

    - **Start (s)**: the moment of the change.
    - **“Volume gauche (%)”** (Left volume) and **“Volume droit (%)”** (Right volume): from 0 to 100.
    - **Length (s)**: used for transitions (see below).
- **Behaviour**: the level changes at once at the start of the block. When two **Sound level** blocks overlap on the same track, the level glides from the first to the second during the overlap (tooltip “overlap = crossfade”).
- The level is a **state**: it stays in force, even in the following scenes, until the next **Sound level** or until you move the sliders. When you move the red line, DreamShow reapplies the last level placed before it.
- The block shows “♪ Level 80 %”, or “♪ Level L 80 % R 60 %” if the two sides differ. In a show imported from ppshow, “· master fade” indicates a level that follows the scene's incoming fade.
- On a track that also carries other blocks, **Stop sound** and **Sound level** are displayed on a lower level.

**Stop sound**, **Silence** and **Sound level** act on the sounds played by DreamShow (sound files). They do not adjust the sound of videos.

## DMX recordings

A .DmxStreaming file (DMX stream, for example produced by an AI or by visualiser mode) or a .bin / .dmx recording dropped from the **Media** list becomes a plum-coloured “▦ file name” block.

- Its initial length is the real length of the recording.
- **Properties…**: only **Start (s)** and **Length (s)** (no fade).
- As long as the red line is inside the block, the recording is replayed at the right position: a .DmxStreaming drives only its own channels, on any universe; a .bin or .dmx entirely replaces the first universe. Pause freezes the picture, a click in the ruler repositions it.

See DMX stream by an AI and Visualiser mode.

## Step by step

### An entrance music that crossfades into a second one

1. Import “entree.mp3” and “valse.mp3” (**Import media…**).
2. Create a “Music” track (**+ track**, then right-click > **Rename the track…**).
3. Drag “entree.mp3” onto “Music” at 0 s; give it a **Fade in (s)** of 3.
4. Drag “valse.mp3” onto the same track, so that it overlaps the last 4 seconds of “entree.mp3”: a cross appears, and the two pieces of music blend into each other.
5. ⏮ then ▶ to listen.

### Cutting the previous scene's music

1. In scene 2, the “Music” track starts a piece of music that lasts longer than the scene.
2. In scene 3, add a track, drag **Stop sound** to 0 s.
3. Double-click it, untick **All tracks**, tick “Music” if a track of that name exists in scene 3 (otherwise, name a track “Music” first). **OK**.
4. On the GO to scene 3, the music stops; the other sounds carry on.

For a gentle ending, use **Silence** instead: placed while a sound of the scene is still playing, it fades all sounds down over its length.

### Lowering the music under a line

1. On a single track, drop a **Sound level** at 10 s (100 %, length 2 s), then a second one at 11 s set to 40 %: they overlap by 1 s, and the level goes from 100 to 40 % in 1 s.
2. Further along the same track, at 30 s, drop a **Sound level** at 40 % (length 3 s), then another at 31 s at 100 %: the level comes back up in 2 s.

### Projecting a video on the back screen

1. Declare the screen in **Venue > Screens…** (for example “Back”).
2. Drag the video onto a “Video” track.
3. **Properties…**: **Screen** = “Back”, **Fade in (s)** 2, **Fade out (s)** 2.
4. The video fades in at the start of the block and goes to black at its end.

## Good to know

- A sound started in a scene is not cut by the GO: provide a **Stop sound** or a **Silence** where it must stop, or adjust the length of its block.
- During a wait block, the scene's sounds and videos pause with the timeline (see Waits).
- The **Mute the sounds** button stops everything, editor and performance alike.
- A robot can speak a recorded text and have it heard on the PA as well: see Humanoid robots.

## See also

- The scene timeline
- Fades and mixes
- Video screens
- Log, Audio and Clock
- Libraries
- Shows, files and projects
- DMX stream by an AI

# Waits (key, time, date, sun)

A wait block stops the scene's timeline at a precise point until a condition is met: the press of a chosen key (**Wait for a key…**), or a moment on the calendar (**Wait for a time, a date, the sun…**): a length of time, a time of day, a date of the year, sunrise or sunset. The second kind of wait restarts by itself, with nobody at the control desk: it is the basis of architectural lighting.

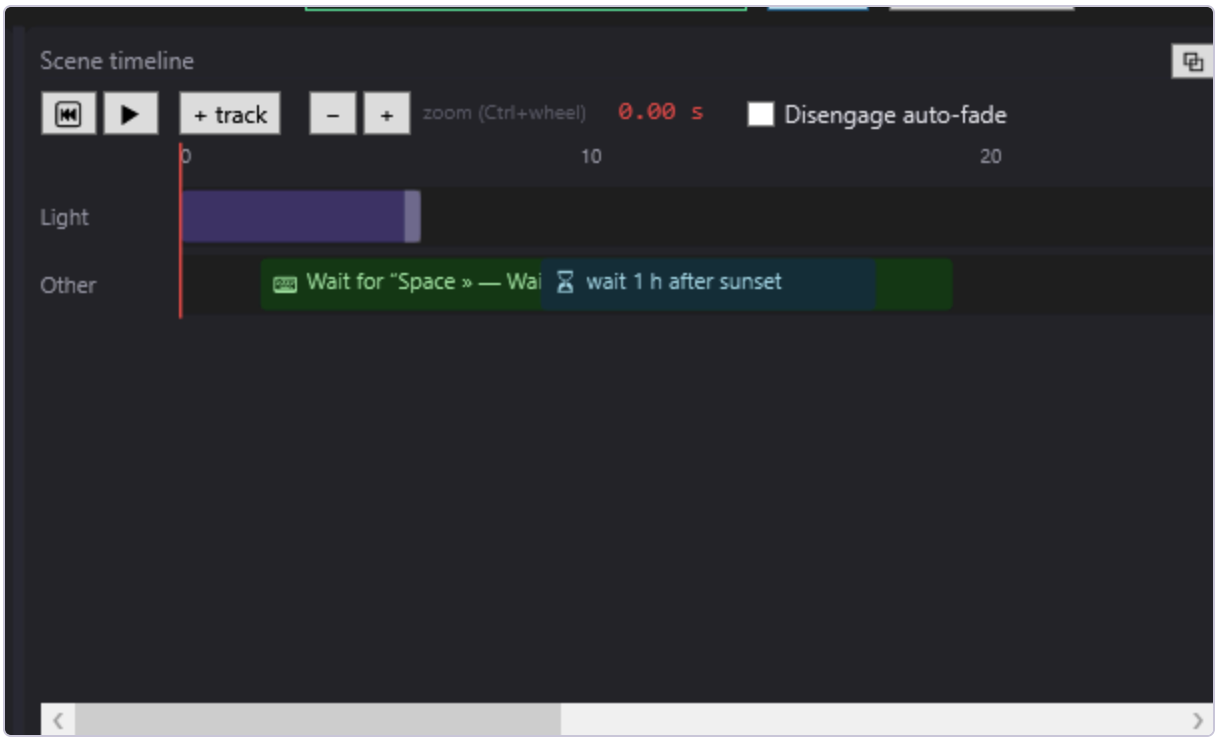
**To open it:** drag **Wait for a key…** or **Wait for a time, a date, the sun…** from the **Others** list (**Libraries** panel) onto a track of the **Scene timeline** panel. The settings window opens at once; double-click the block to reopen it.

## What happens during a wait

When the red line reaches a wait block:

- the **scene's timeline stops** at that point: the following blocks do not start;
- the light **stays as it is** at that instant (final state, plus the blocks in progress, frozen at their position; a dynamic effect stops on the spot); the DMX output carries on normally;
- the scene's **sounds and videos are paused**, and resume when the wait ends;
- any **parallel sequences** in progress also freeze until the wait ends;
- when the wait ends, the timeline carries on exactly where it stopped.

A wait is an instantaneous event: only its start counts.



## Wait for a key...

Stops the scene until the operator presses a specific key on the keyboard. Useful for waiting for a signal from the stage without leaving the scene (end of applause, an actor's entrance).

### The “Wait for a key” window

- **Start (s)**: the moment when the scene stops.
- **Key**: the key awaited (**Space** by default). To change it, click **Set...** (the button shows “Press a key...”), then press the key you want. This deliberate capture avoids recording Tab, Enter or Esc by mistake while moving around the window. Keys are displayed by name: **Space**, **Enter**, **Esc**, **Backspace**, **Page** ↓, **Page** ↑, ← **Left**, **Right** →; the others under their Windows name (for example “F5”, “A”, “D1” for the 1 key of the main keyboard).
- **Message while waiting:** text displayed during the wait (optional), for example “Wait for the end of the applause”.
- **OK** / **Cancel**.

The block shows “⌨ Wait for “Space” — start of the message”.

### During the wait

- **In the cue window**, the message area shows your message in large type, with the key (“⌨ “Space””); with no message, it shows “⌨ Press “Space””.
- **In the editor**, the red line's position is replaced by “⌨ wait “message” → “Space””.
- The wait is released when **that** key is pressed, with the DreamShow window (editor or cue window) in focus. Any other key does nothing.
- The awaited key is served first: if it is also the GO key, it releases the wait **without** triggering a GO.
- An OSC remote can also release the wait with the command `/dreamshow/key` followed by the Windows name of the key (Space, F5...). See The OSC remote.
- A presentation remote usually sends **Page** ↓: choose that key to release the wait with the remote. See The remote (GO keys).

## Wait for a time, a date, the sun...

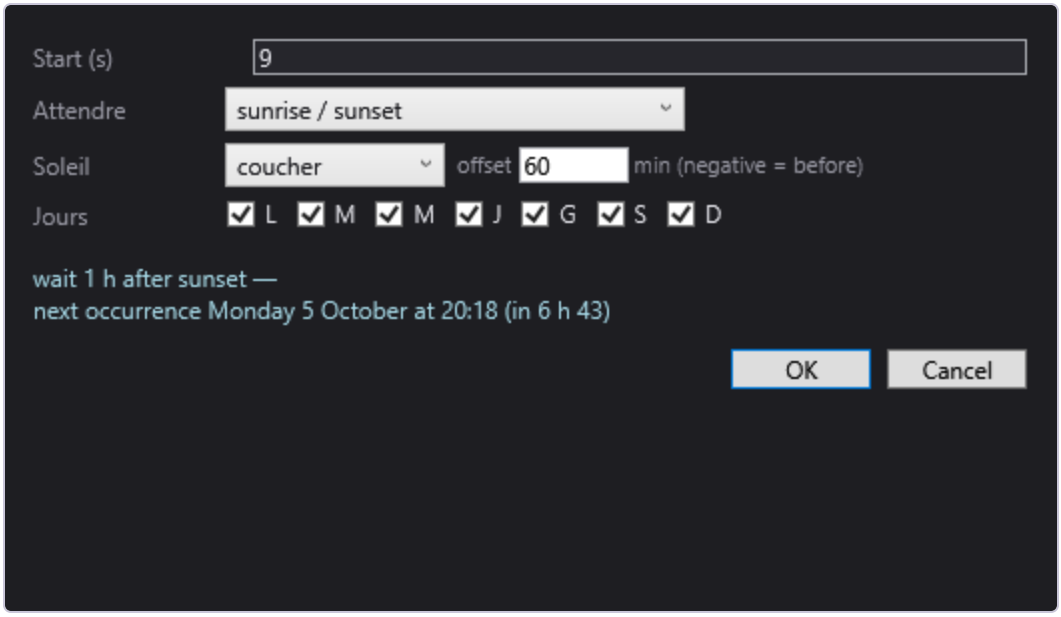
Stops the scene until a given moment, then **restarts by itself**, with no GO. Block tooltip: “The show resumes by itself when the time comes; GO overrides it.”

### The “Wait for a time, a date, the sun” window

- **Start (s)**: the moment when the scene stops.
- **“Attendre”** (Wait for): the kind of rule. Depending on the choice, only one of the following lines is displayed.

    - **a length of time**: **Length** line. The waiting time, counted from the moment the red line reaches the block. Accepted notations: “90” or “90 s” (seconds), “5 min” (or “5 m”, “5 mn”), “2 h”, “1h30”, “1:30” (= 1 h 30), “01:30:00”. Suggested value: “1 h”.
    - **a time of day** (default choice on creation): **“Heure”** (Time) line, in the format “hh:mm” (“18:30”; “18h30” and “18h” are also accepted). Suggested value: 18:30.
    - **a date (every year)**: **“Date”** line, in the format “dd/mm [hh:mm]” (“jj/mm [hh:mm]” on screen), for example “24/12 18:00”. The time is optional (no time: midnight, at the start of the day). The rule comes back every year. 29/02 is only reached in leap years. Suggested value: “24/12 18:00”.
    - **sunrise / sunset**: **“Soleil”** (Sun) line, with the **sunset** / **sunrise** list and the **offset** in minutes (“min (negative = before)”): 60 = one hour after, -15 = a quarter of an hour before. Suggested value: 60.
- **“Jours”** (Days) (for **a time of day** and **sunrise / sunset** only): **L M M J V S D** boxes (Monday to Sunday, French initials). Untick the days on which the rule must not apply: for example everything except **S** and **D** for “weekdays only”. All ticked, or all unticked, means “every day”.
- **The preview** (blue text at the bottom): the rule in plain words and its **next occurrence**, for example “wait until 18:30 (except Sat, Sun) — next occurrence Monday 6 October at 18:30 (in 3 h 12)”. It updates at every change. If the rule is impossible (badly written time…), it shows “rule impossible to satisfy: the block would be ignored when reached”.
- **OK** / **Cancel**.

The block shows the rule: “ ⏳ wait until 18:30 (except Sat, Sun)”, “ ⏳ wait 1 h after sunset”, “ ⏳ wait for 24 December 18:00 (every year)”, “ ⏳ wait 5 min”…




## The next occurrence

The deadline is computed **at the moment the red line reaches the block**, never before: a show prepared at noon does wait for “the next 18:30” when it passes over the block at midnight.

And it is always the **next** occurrence, never one that has already gone by. If the “wait until 18:30” block is reached at 18:31, or if the computer restarts at 23:00, it waits for 18:30 **the next day** (respecting the ticked days). This fixed rule makes unattended installations predictable.

### Sunrise and sunset

They are computed for the venue's location, every day, without the Internet (official times, accurate to within a minute; daylight saving time is handled by Windows).

Set the location in **Venue > Fixtures, addresses, positions…**, fields **Location (°) — latitude** and **longitude** (“north and east positive — used to compute sunrise and sunset”). Example: Paris = 48.85 and 2.35. With no setting, the venue has coordinates in the south of France (44.126 and 4.082). See Fixtures, addresses, positions.

### During the wait

- **In the editor**, the red line's position is replaced by the rule and the deadline: “ ⏳ wait until 18:30 → Mon 18:30 (in 3 h 12)”.
- The **Log** notes the start (“Waiting: … → resumes … (in …).”) and the end (“Wait finished (…) — resuming.”).
- At the deadline, the engine restarts by itself, even if nobody is there, even if a window is open.
- An impossible rule never blocks the show: the block is ignored when reached, with a warning in the **Log** (“Wait block impossible to satisfy (…): it is ignored.”).

## GO, scene changes and waits

- A **GO** leaves the scene for the next one, and cancels the wait in progress. But GO is only allowed when the scene's timeline has finished (or the scene is on **Loop**): if the wait block is the last element of the timeline, GO is possible during the wait; if there are other blocks after it, the GO button stays greyed out until the wait ends.
- Entering another scene (GO, jump, choosing a scene in the editor) always cancels the wait in progress.
- In the editor, ▶ cancels the wait and restarts playback without waiting for the deadline or the key: the red line's position is displayed again in place of the wait text.

## Waits and automatic chaining

A scene set to **Auto** or **Chained** counts its delay in timeline time, and that time stops during the wait:

- **Auto**: place the wait **before** the delay. Example: wait at 0 s, **Auto** with a delay of 2 s → the scene waits, then moves on to the next one 2 s after the wait ends, with its fade.

- **Chained**: the scene leaves at the end of its last block; there must therefore be at least one block that ends **after** the wait.

Never place the wait **at the very end** of the timeline of a **Chained** scene (nor at the very moment of the delay of an **Auto** scene): the scene would leave for the next one at once, without waiting, and the next one would stay stopped at 0 s until a GO.

## In a parallel sequence

A parallel sequence cannot wait: any wait blocks (key or calendar) it meets are ignored, with a warning in the **Log**. On the other hand, a wait in the main show also freezes the parallel sequences until it ends. See Parallel sequences.

## Step by step

### Waiting for the end of the applause

1. In the “Bows” scene, drop **Wait for a key…** at 4 s.
2. Keep **Space**, write the message “Restart at the end of the applause”. **OK**.
3. Drop after it, at 5 s, the lighting block for the curtain call.
4. In performance, the scene stops at 4 s and the cue window shows the message; Space restarts the rest.

### Switching on a facade a quarter of an hour before sunset, on weekdays

1. Set the venue's latitude and longitude (**Venue > Fixtures, addresses, positions…**).
2. Scene 1 “Evening wait”: empty timeline, final state off. Drop **Wait for a time, a date, the sun…** at 0 s.
3. Choose **sunrise / sunset**, **sunset**, offset **-15**, untick **S** and **D**. Check the preview. **OK**.
4. In **Scene settings**, **Trigger: Auto**, delay 2; **Fade to the next (s):** 10.
5. Scene 2 “Facade”: the lighting you want.
6. When evening comes, scene 1 is released, then the facade lights up over 10 s.

The complete recipe, with switching off and starting over every day, is in Architectural lighting.

### A timed pause

1. Drop **Wait for a time, a date, the sun…**, choose **a length of time**, write “20 min”.
2. The show restarts by itself 20 minutes after reaching the block. If the block is the last element of the scene, a GO lets you move on to the next scene without waiting for the end.

## Good to know

- During a wait, the picture is frozen: a dynamic effect, a trajectory or a pixel mapping stops on the spot. For lighting that must stay animated during a long wait, prefer a scene on **Loop** with a **max length** (see Architectural lighting).
- The cue window does not show the countdown of a time-of-day wait: it is visible in the editor (red line position) and in the **Log**.
- The export for a standalone player computes the show in advance: time-of-day waits are ignored there. The player has its own time schedule, see The standalone player.
- Date rules have no day filter; neither do length-of-time rules.
- Imported shows may contain an old " ⏸ Pause" block (window title "Pause (wait)"): it stops the timeline until ▶ in the editor or a GO. It is no longer offered in the **Others** list; replace it with **Wait for a key…**.

## See also

- Wait for a spoken line (voice sync)
- The scene timeline
- Architectural lighting
- Scene settings
- Jumps and scene labels
- Show mode and the cue window
- The remote (GO keys)
- Fixtures, addresses, positions

# Wait for a spoken line (voice sync)

The **Wait for a spoken line…** block stops the timeline until a performer has said a given line into the microphone. DreamShow recognises the voice by itself, without internet: a dialogue between a performer and a robot, or a series of effects cued on a text, can run with nobody at the desk.

**To open it:** drag **Wait for a spoken line…** from the **Others** list (**Libraries** panel) onto a track of the **Scene timeline** panel. Its settings window opens at once; double-click the block to open it again.

## How it works

- During the block, and **only during the block**, DreamShow listens to the microphone. As soon as the line is recognised, the timeline moves on.
- Outside the blocks nothing is listened to: a voice played by the show itself (a robot's line, a sound) cannot trigger anything, provided it is placed **after** the block.
- For a dialogue: one **Wait for a spoken line…** block for each of the performer's lines, followed by the blocks of the answer (**Robot** block “Dire” (Say), sound, light…), then the block of the next line.

Like the other waits, the block freezes the light and pauses the scene's sounds and videos while listening (see Waits (key, time, date, sun)).

## The block's settings

- **Expected line**: the text, as it will be said. If the show has a teleprompter text, the list **Take a line from the teleprompter text…** copies it.
- **Accept a line said approximately** (ticked by default):
  - if the **end** of the line is said as written, the show carries on at once (usually less than half a second after the last word);
  - otherwise the **gist** is enough: about half of the important words, in order, including one from the end of the line. The show then carries on after **1.5 s of silence**: a mere pause in the middle of the sentence does not set it off.
  - Unticked: only the exact end counts.
- **Fallback key phrase** (optional): a phrase the performer can say if the line went wrong or was not recognised, for example « voilà voilà ». It moves on at once. It must be said exactly, and its words must not appear in the line (the window warns otherwise). A new block takes the key phrase already used in the show.
- **Fallback delay (s)**: if the line is not recognised within this delay, the show carries on anyway and notes it in the log. Empty: it waits with no limit.

- **Microphone**: the microphone listened to, the same for every block on this PC (setting kept on the PC, not in the show).
- **Spoken language**: the actors' language (French, English, German, Spanish or Italian), the same for the whole show, blocks and teleprompter by voice. French comes with DreamShow; another language is downloaded once (40 to 50 MB) as soon as it is chosen or a show using it is opened: do it while preparing, with Internet. After that everything runs on the PC, without Internet.
- **Start (s)**: the block's position on the timeline.

On the timeline, the block shows 🎤, the beginning of the line and its fallback delay.

## During the show

In the cue window, while listening, the expected line is shown under the scene name with what DreamShow hears (“listening…”, then “heard: …”) and the fallback countdown.

If the fallback key phrase is said, the show carries on and the log notes it as a warning: the lines to work on can be found afterwards.

**GO** releases the wait at any time: the operator can always take over. If the microphone cannot be found, the window says so (“cannot listen”) and the line waits for GO or the fallback delay.

The log (category **lines**) notes each listening, each recognised line (with the share of words heard, and “approximate” where relevant) and each fallback delay that ran out.

## Tips

- **The microphone matters most**: a wireless headset microphone, as close to the mouth as possible, gives the best recognition. An ambient microphone hears the audience and the loudspeakers.
- **Take care of line endings**: they are what sets the show off without waiting. An ending with a distinct word is recognised better than one in “yes” or “eh”.
- Very short lines (“Oui.”, “Ah?”) must be said as written: plan a fallback delay.
- The text may contain stage directions in parentheses or brackets: they are not expected.
- Rehearse with the real microphone and watch the “heard: …” line of the cue window: it shows what DreamShow understands.

## Limits

- Five languages recognised: French, English, German, Spanish, Italian. One language per show.
- One voice is expected at a time, on one microphone.
- A skipped line is not caught up: the block waits for its line, then the fallback delay or GO.

## See also

- Waits (key, time, date, sun)
- Humanoid robots
- The teleprompter
- Show mode and the cue window
- The scene timeline

# Jumps and scene labels

A **jump** takes the cue list from one scene to another at the moment the red line reaches it, without a GO: to go back, skip a passage, or loop several scenes. A scene's **label** is a short, stable name that lets jumps (and parallel sequences) always target the right scene, even when scenes are inserted or moved. This page also presents the two ways of making a loop.

**To open it:** drag **Go to scene…** from the **Others** list (**Libraries** panel) onto a track of the **Scene timeline** panel. The label is set in **Scene settings**, in the **Label:** field.

## The “Go to scene…” block

### Creating it

1. Drag **Go to scene…** from the **Others** list onto a track, at the moment of the jump.
2. A small **Go to scene** window opens: “Jump to scene:”, with the list of scenes (“3 · Final”). The scene that follows the selected scene is suggested.
3. Choose the scene and click **OK**. Closing the window without **OK** cancels: no block is created.

The block (teal) shows the target scene like the **Scenes** list: “→ 3 · Final [final]” (number, name, and label in square brackets if there is one).

### Setting it

Double-clicking the block (or right-click > **Properties…**) opens the **Go to scene** window:

- **Start (s)**: moment of the jump.
- **Target scene**: the scene targeted.

It is an instantaneous event: only its start matters.

### What happens at the jump

- As soon as the red line reaches the block, the current scene is **left immediately** and the target scene starts at 0 s.
- The jump is **instantaneous, with no fade**: the light switches at once to the final state of the target scene. To soften it, give a fade in to the blocks at the start of the target scene, or use an automatic chain (**Auto** or **Chained**, which apply the **Fade to the next**) when the aim is to go to the next scene.
- Blocks placed **after** the jump in the scene being left are never played.

- As with a GO: the videos of the scene being left stop, the sounds carry on.
- A jump to the scene itself makes it start again at 0 s.
- In the editor and in Show mode, the selection in the **Scenes** panel follows the target scene (in the editor, when the **LIVE** button is lit).

### Broken jump

If the target scene no longer exists, the block turns red with ⚠ and shows "→ scene? (not found)". When it is reached, the jump is ignored and the **Log** says so ("Jump to "…": scene not found — ignored."). Double-click the block to choose another scene.

## A scene's label

The **Label:** field is next to the scene's name, in **Scene settings**. Tooltip: "Short, stable name of the scene (e.g. rain): jumps and parallel sequences target it by this name, even when scenes are inserted or moved".

- Write a short word: "rain", "final", "waiting". Capital letters do not matter. An "@" typed at the start is removed.
- The label appears in square brackets in the **Scenes** list and in the scene choice lists ("5 · Storm [storm]").
- Leave the field empty for a scene that is the target of no jump.

### Why it matters

A jump keeps hold of its target scene in two ways:

- **by its label**, if the scene had one when you confirmed the block: the jump always targets that scene, wherever it is in the list;
- **by its number** otherwise. DreamShow updates these numbers when you add (**+**), duplicate (⎘) or delete (**−**) a scene. But **not** when you move a scene with ↑ or ↓: the jump then targets the scene that has taken that place.

So give a label to every scene targeted by a jump or a parallel sequence.

### Switching an existing jump to the label

A jump created before the target scene had a label stays attached to the number. To attach it to the label:

1. Give the target scene a label.
2. Double-click the jump block, check the **Target scene**, then **OK**: the block now records the label.

### Renaming, deleting, duplicating

- **Changing a label**: the jumps that used the old label turn red (⚠). Reopen them and confirm them again.
- **Deleting a target scene**: a jump by label turns red; for a jump by number, DreamShow warns ("N "go to scene" pointed at the deleted scene: to be fixed (shown in red ⚠ on the timeline).").
- **Duplicating a scene** (⎘) also copies its label: two scenes then carry the same one. The **Log** flags it ("Label "…" used by several scenes: the first one (…) is kept.") and jumps target the first one. Change or clear the copy's label.

## Loops

### A scene that loops on itself

In **Scene settings**:

- **Loop**: "The timeline loops automatically until GO (or until the maximum length)". When the red line reaches the end of the last block, the timeline starts again from 0: the blocks replay, the sounds and videos restart (on each turn, the sounds and videos in progress are stopped first).
- **max length (s):** "Empty = loops until GO; otherwise the scene chains to the next one after this total time". The time is counted over all the turns; when it runs out, the scene moves on to the next one with its **Fade to the next**, even in the middle of a turn.

GO is always allowed on a looping scene: it leaves it at any time. **Chained** makes no sense on a looping scene (it never ends): only GO and the max length take it out. The cue window shows the current turn ("↻ loop 3 (max 600 s)"). See Scene settings.

### A loop of several scenes

To loop a series of scenes:

1. Give a label to the first scene of the cycle (for example "cycle").
2. Set the scenes of the cycle to **Auto** or **Chained**, so that they follow on by themselves.
3. In the last scene of the cycle, place **Go to scene…** at the end of its timeline, targeting "cycle".

The cue list then runs endlessly: scene 1, 2, 3, jump to 1…

Such a loop cannot be left with GO: GO is only allowed at the end of the timeline, and the jump goes off at that moment. To get out, jump directly to another scene from a remote (web, OSC, MIDI), or reset the cue list. If the cycle must be left on GO, prefer a single scene set to **Loop**.

## Step by step

### Replaying a passage until the director's signal

1. Scene 4 “Rehearsal”: set **Loop**, leave **max length (s):** empty.
2. During the show, the scene loops; on the signal, GO moves to scene 5, with scene 4's fade.

### Skipping a scene on certain evenings

1. Scene 6 “Short scene”: place **Go to scene…** at 0 s targeting scene 9 (give it the label “after-scene”).
2. On the evenings when the full scene is played, delete the jump, or move it past the end of the timeline.

### A shop-window animation that runs endlessly

1. Scenes 1 “Red”, 2 “Gold”, 3 “White”: **Auto** trigger (for example 30 s), **Fade to the next (s):** 5.
2. Label “window” on scene 1.
3. In scene 3, place **Go to scene…** targeting “window” at 30 s.
4. The Red → Gold → White cycle runs indefinitely; the return from White to Red is instantaneous (it is a jump). For a fading return, add to scene 1 a light block with a fade in.

## Good to know

- Placed in a scene played by a parallel sequence, a jump makes the sequence itself jump, without touching the main cue list; that jump has no fade either (see Parallel sequences).
- The end of the show (“END OF SHOW” in the cue window and the **Log**) is noted when the cue list arrives on the last scene of the list by a GO or an automatic chain.
- Export for a standalone player calculates the show in advance, scene after scene: a jump to a scene that has already been played ends the export after a single cycle, which the player replays in a loop from the start of the clip. See The standalone player.

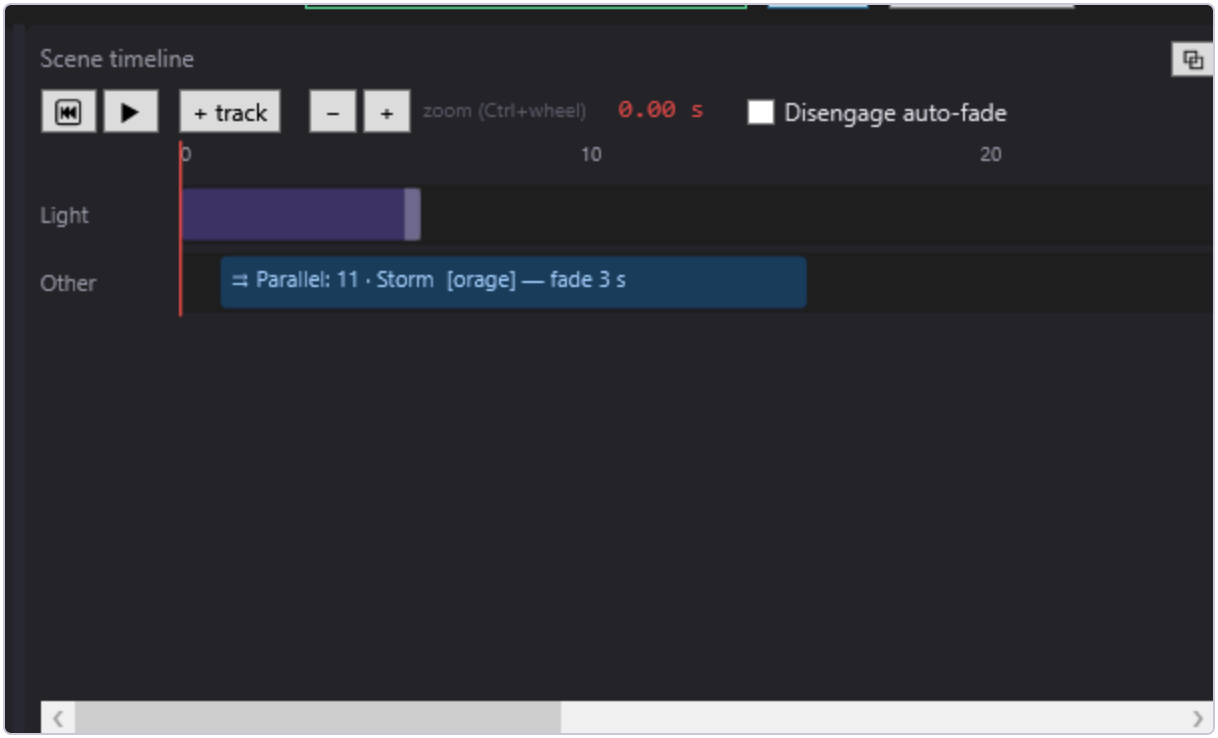
## See also

- The scene timeline
- Scene settings
- The Scenes panel
- Parallel sequences
- Architectural lighting
- Fades and mixes
- The web remote

# Parallel sequences

A parallel sequence is a second playback that runs **at the same time** as the cue list: it plays a scene, then the following ones according to their automatic chains, while you keep advancing the cue list with GO. A typical example: a storm (lightning, thunder, rain) that starts during one scene and carries on through the next three scenes, whatever happens in the cue list, until a scene fades it out.

**To open it:** drag **Start a parallel sequence…** or **End the parallel sequence…** from the **Others** list (**Libraries** panel) onto a track of the **Scene timeline** panel. The settings window opens at once; double-click the block to reopen it.




## The principle

- The scenes played by a parallel sequence are **ordinary scenes** of the show: you create and set them like the others (final state, timeline, sounds…).
- The **Start a parallel sequence…** block, placed in a scene of the cue list, launches the sequence from a chosen scene (its **starting scene**).
- The sequence plays this scene, then moves on to the following scenes **by itself**, according to their automatic triggers: **Auto**, **Chained**, **Loop** (and its max length), and the **Go to scene…** jumps placed in its scenes. A scene set to **Manual GO** stays in place: **GO only advances the main cue list**.
- The **End the parallel sequence…** block stops it, with a fade.

- A sequence is identified by its starting scene. Several sequences with different starting scenes can run at the same time.

Give a **label** to the starting scene (the **Label:** field of **Scene settings**, for example “storm”): the blocks find it again even if you insert or move scenes. See Jumps and scene labels.

## Start a parallel sequence…

### The “Start a parallel sequence” window

- **Start (s)**: the moment the sequence starts, in the timeline of the scene that carries the block.
- **Starting scene**: the scene the sequence begins with (list “5 · Storm [storm]”). On creation, DreamShow suggests the scene that follows the selected scene.
- **Fade in (s)**: the sequence's rise time (1 s suggested). 0 = it appears all at once.
- Help text: “The sequence plays this scene and then the following ones by their automatic triggers, at the same time as the main show. Go only moves the main show forward. Give the scene a label so that it stays targeted if the order changes.”
- **OK** / **Cancel** (cancelling at creation = no block).

The block shows “⇉ Parallèle : 5 · Storm [storm] — fondu 3 s” (the block text itself stays in French: Parallèle = parallel, fondu = fade).

If the same sequence is already running (same starting scene), the block **restarts it from its first scene** (its sounds and videos in progress are stopped), with the new fade in.

## End the parallel sequence…

### The “End the parallel sequence” window

- **Start (s)**: the moment the sequence begins to fade out.
- **Sequence to end**:
  - **(the parallel sequence this block is in)**: to be used when the block is placed **in a scene of the sequence** itself; the sequence ends by itself;
  - a scene from the list: to be used when the block is placed **in the cue list**; it ends the sequence started from that scene.
- **Fade out (s)**: the time during which the sequence fades away (2 s suggested). 0 = it disappears all at once.
- Help text: “Placed in the parallel sequence, this block ends it; placed in the main show, it ends the sequence started from the chosen scene.”

The block shows “⇥ Fin parallèle : cette séquence — fondu 2 s” or “⇥ Fin parallèle : 5 · Storm [storm] — fondu 6 s” (“Fin parallèle” = end of parallel; “cette séquence” = this sequence).

At the end of the fade out, the sequence disappears and its sounds and videos are stopped (abruptly: the fade out concerns the light).

If the targeted sequence is not running, the block does nothing. A block set to **(the parallel sequence this block is in)** but placed in the cue list does nothing either, and the **Log** flags it: ""End the parallel sequence" with no sequence to end: place it in the parallel sequence, or name its starting scene."

The ⏮ **Reset (Ready)** of the cue window stops all parallel sequences. A GO, a jump or a scene change in the cue list does not stop them.

## What a sequence plays

In each of its scenes, a sequence plays:

- the scene's **final state**, but only for what it sets (fixtures absent from its final state are not set to zero: they stay with the cue list);
- its **light and effect blocks**, **trajectories**, **dynamic effects**, **pixel mappings** and **Blackout** (a Blackout there only turns off the sequence itself);
- its **sounds** and **videos** (they are noted in the **Log**: ""storm": playing tonnerre.wav"), the **Backstage display…**, **Robot** and **Mask** blocks, the blocks that start or end other sequences, and the **EXEC=** extra command.

It moves from one scene to the next with the **Fade to the next** of the scene being left; a **Go to scene…** jump there is instantaneous.

What a sequence **ignores** (with a warning in the **Log**, once per sequence): waits (**Wait for a key…**, **Wait for a time, a date, the sun…**), the old Pause block, and the extra commands that take over the whole show (CHAIN=, REPLAYDMX=, AMBILIGHT=). Message: "Parallel sequence "…": "…" ignored (only the main show can wait or chain)."

Also not played in a sequence: DMX recordings (▦ blocks). And two light blocks that overlap there only make an intensity fade (no colour or position mix).

## Mixing with the cue list

The sequence mixes with the cue list according to the usual lighting-control rules:

- **Intensity and colour mixing** (dimmer, red, green, blue, white…; channel by channel): **the highest wins** ("HTP" mode, for *highest takes precedence*). The sequence can therefore only add light: it never darkens the cue list. Example: the cue list sets the front light to red, the sequence sets it to blue on the same fixtures → the front light becomes magenta.
- **All the other channels** (position, zoom, colour and gobo wheels, shutter…): **the sequence wins** on the channels it sets ("LTP" mode, for *latest takes precedence*). During the sequence's

fade in or fade out, a position slides between the cue list and the sequence; a wheel or a shutter switches halfway through the fade.
- The **general blackout** (the cue window's **BLACKOUT** button) and the **manual controls** win over both.
- A **Blackout** block in the cue list does not turn off the parallel sequences.

To avoid surprises, have a sequence play fixtures of its own (for example the backlights for a storm), or accept that the position of those fixtures comes from the sequence as long as it is running.

## Where to see the running sequences

- **In the editor**: the information line at the top of the window (to the right of the show's name) adds "· ⇉ en parallèle : storm" ("en parallèle" = in parallel; the label of the starting scene, or its name).
- **In the cue window**: the top line adds "⇉ en parallèle : storm".
- **In the Log**: "Parallel sequence "storm" started (scene 5), fade 3 s", then "Parallel sequence "storm" ended, fade 6 s".

## Pauses and the editor

A parallel sequence follows the engine's clock: **it freezes whenever the cue list is paused** (the editor's ⏸ button, a stop at the end of the timeline in the editor, a wait block in the cue list), and restarts with it.

In the editor, a scene set to **Manual GO** stops playback at the end of its timeline: the parallel sequence therefore stops too. To watch a sequence evolve for a long time, switch to **Show mode** (the cue list runs without stopping), or set the cue list's scene to **Loop** for the duration of the rehearsal.

The **Disengage auto-fade** box also blocks the sequences' automatic chains (a scene set to **Loop** keeps looping).

## Complete example: a storm that carries on while the cue list moves forward

The show: a peaceful village, the storm breaks during scene 1, lasts through the argument in scene 2, and calms down at the start of scene 3. The groups "Front", "Cyc" and "Backlights" exist in the venue; the backlights are reserved for the storm. The sounds "pluie.wav" (rain) and "tonnerre.wav" (thunder) are in the show's media.

### The scenes

| No. | Name | Label | Trigger | Role |
|---|---|---|---|---|
| 1 | Village | | Manual GO | starts the storm |
| 2 | Argument | | Manual GO | the storm carries on |
| 3 | Lull | | Manual GO | ends the storm |
| 4 | Bows | | Manual GO | last scene played by the cue list |
| 5 | Storm | storm | Manual GO, **Loop** | the parallel sequence |

### Step by step

1. **Create scene 5 “Storm”** (after “Bows”):
   - **Label:** “storm”; tick **Loop**, leave **max length (s):** empty.
   - Final state: empty (the storm only lights with its blocks).
   - “Lightning” track: drag “Backlights” from **Groups** at 2 s. If “Backlights” is not set in any scene, the block is white at 100 % by default; otherwise, set it to white at 100 % with **Replace with the current selection + settings** (see Light and effect blocks). Set its **Length (s)** to 0.2. Drop two other flashes in the same way, at 2.4 s and 7.5 s.
   - “Sounds” track: “tonnerre.wav” at 2.6 s; on a “Rain” track, “pluie.wav” from 0 to 12 s with **Fade in (s)** and **Fade out (s)** of 0.5.
   - The timeline lasts 12 s (end of the rain sound): the scene loops every 12 s.
2. **Scene 1 “Village”**: final state Front 70 % warm, Cyc sky blue. Drag **Start a parallel sequence…** onto a track at 15 s; **Starting scene**: “5 · Storm [storm]”; **Fade in (s)**: 3.
3. **Scene 2 “Argument”**: its usual lighting. Nothing to do for the storm: it carries on.
4. **Scene 3 “Lull”**: drag **End the parallel sequence…** at 0 s; **Sequence to end**: “5 · Storm [storm]”; **Fade out (s)**: 6.
5. **Try it in Show mode**: click **Show mode**, then GO. At 15 s in “Village”, the top line shows “⇉ en parallèle : storm”, the lightning and the rain come up over 3 s. GO to “Argument”: the front light changes, the storm carries on (it loops every 12 s). GO to “Lull”: the storm fades away over 6 s, then the rain stops.

During the storm, the backlights come from the sequence; the front light and the cyc stay those of the cue list (the sequence does not set those fixtures).

### Variant: a sequence that ends by itself

For a two-part fireworks display: scene “Fireworks 1” (label “fireworks”, **Chained**) then “Fireworks 2”; at the end of the timeline of “Fireworks 2”, place **End the parallel sequence…** set to **(the parallel sequence this block is in)**. The sequence plays “Fireworks 1”, chains on to “Fireworks 2”, then fades away.

## Good to know

- A sequence's scenes are part of the show's list: the cue list can also reach them with GO. Put them **after** the last scene of the cue list, and do not GO beyond it. The “END OF SHOW” message in the cue window only appears on arriving at the last scene of the list.
- To set up a sequence scene, select it in the **Scenes** panel: the editor then plays it in the main cue list, like any scene.
- A parallel sequence can itself start or end other sequences.
- If the sequence arrives at the last scene of the list, it stays there.

## See also

- The scene timeline
- Jumps and scene labels
- Waits (key, time, date, sun)
- Fades and mixes
- Show mode and the cue window
- Scene settings

# Architectural lighting

A facade, a hall, a shop window or a garden is lit every evening, with nobody at the control desk: switching on at sunset, looping animations, moving to fixed lighting, switching off at a set time, then everything starts again the next day. This page explains how to build this programme with DreamShow's wait blocks, loops and jumps.

**To open it:** there is no dedicated window: it is an ordinary show, built in the **Scenes** panel and in the **Scene timeline** with the items of the **Others** list (**Wait for a time, a date, the sun…**, **Go to scene…**). It then runs in **Show mode**.

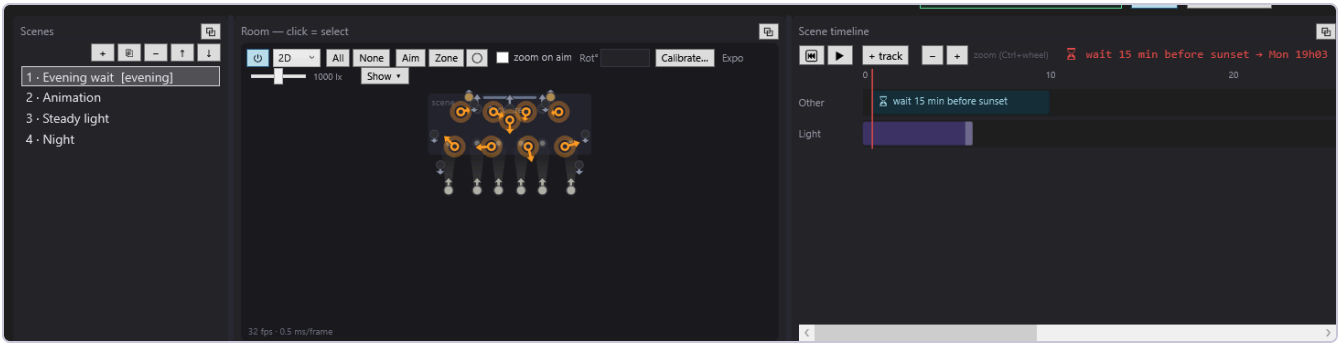
## The principle

The show goes round in a circle, over four scenes:

| No. | Scene | Label | Contents | Trigger |
|---|---|---|---|---|
| 1 | Evening wait | evening | blackout; “sunset” wait block at 0 s | **Auto**, 2 s |
| 2 | Animation | | lighting animations | **Loop** with **max length** |
| 3 | Fixed lighting | | fixed lighting; “23:30” wait block at 0 s | **Auto**, 2 s |
| 4 | Night | | blackout; **Go to scene…** block to “evening” | manual GO |

1. Scene 1 waits for the next sunset, then moves on to scene 2 with a fade.
2. Scene 2 loops its animations for the chosen length, then moves on to scene 3.
3. Scene 3 holds fixed lighting until 23:30, then moves on to scene 4 with a fade (the switch-off).
4. Scene 4 jumps to scene 1, which waits for **the next day's** sunset (a wait always aims at the next occurrence).

These mechanisms are detailed in Waits and Jumps and scene labels.




## Before you start

- **The venue's location**: set **Location (°) — latitude** and **longitude** in **Venue > Fixtures, addresses, positions…** (north and east positive). Without it, sunset is computed for other coordinates. See Fixtures, addresses, positions.
- **DMX output** to the fixtures: see DMX connections.
- **The computer** must stay on, with no sleep mode, with DreamShow open on the show in **Show mode**. To do without a computer, see the section on the standalone player below.

## Step by step

### 1. The evening wait scene

1. Create scene 1, name it “Evening wait”, **Label:** “evening”.
2. Leave its final state empty: everything is off.
3. Drag **Wait for a time, a date, the sun…** onto a track, at 0 s.
4. In the window: **“Attendre”** (Wait for) = **sunrise / sunset**, **sunset**, offset **-15** (a quarter of an hour before). Leave all the **“Jours”** (Days) ticked (or untick those on which the facade must stay off). Check the preview (“next occurrence …”), then **OK**.
5. In **Scene settings**: **Trigger:** = **Auto**, delay **2**; **Fade to the next (s):** = **20** (switching on will take 20 s).

The **Auto** delay must be greater than the position of the wait block: the scene waits, then restarts 2 s after the deadline. Do not set this scene to **Chained**: with the wait as its only element, it would leave without waiting.

### 2. The animation scene

1. Create scene 2 “Animation”. Set its final state (the facade's basic lighting).
2. Build an animated timeline, for example:
    - a “Colours” track with three overlapping light blocks (mix) to cycle the colours over 2 minutes (see Fades and mixes);
    - a slow **Dynamic effect…** on the cornice fixtures (see Dynamic effects);
    - or a **Pixel mapping…** on LED bars (see Pixel mapping).

3. Tick **Loop**. In **max length (s):**, enter the total length of the animation, for example **5400** (1 h 30).
4. **Fade to the next (s):** = **10**.

Choose the max length so that the animation ends **before the switch-off time, even on the longest day of the year**: if scene 3 is reached after 23:30, its wait would aim at 23:30 the next day, and the facade would stay lit all night and all day. Example: sunset around 21:55 at the end of June, switch-on 15 min before, that is around 21:40 → the animation must last less than 1 h 50.

### 3. The fixed lighting scene

1. Create scene 3 “Fixed lighting”, with the final state you want for the end of the evening.
2. Drag **Wait for a time, a date, the sun…** to 0 s: **a time of day**, **23:30**. **OK**.
3. **Trigger:** = **Auto**, delay **2**; **Fade to the next (s):** = **60** (slow switch-off over one minute).

During a wait, the picture is frozen: this is why this scene's lighting is fixed, and the animations are in scene 2.

### 4. The night scene and the return to the start

1. Create scene 4 “Night”, empty final state (blackout).
2. Drag **Go to scene…** to 5 s, target “1 · Evening wait [evening]”.
3. Leave the trigger on **Manual GO**.

On the jump, scene 1 starts again and computes the **next** sunset: the next day's.

### 5. Starting it up

1. Check that **Disengage auto-fade** is **unticked** (otherwise nothing chains).
2. Save the show (Ctrl+S).
3. Click **Show mode**, then **GO**: the cue list enters scene 1 and waits.
4. Check in the editor the red line's position (“ ⏳ wait 15 min before sunset → Mon 18:47 (in 2 h 05)”) or in the **Log** (“Waiting: … → resumes …”). The cue window does not show this countdown.

If you start one evening after the switch-on time, scene 1 would wait for the next day: press **GO** a second time, and the cue list goes straight to scene 2. Likewise, a **GO** on scene 3 switches off without waiting for 23:30.

## Variants

- **Weekdays only**: in scene 1's wait block, untick **S** and **D**. On Friday evening, after the switch-off, scene 1 waits for Monday's sunset.
- **No animation**: delete scene 2; scene 1 leads directly to scene 3's fixed lighting.

- **Switching off at sunrise**: in scene 3, set the wait to **sunrise / sunset**, **sunrise**, offset 0 (or negative to switch off earlier).
- **Switching on at a fixed time** rather than at sunset: in scene 1, set the wait to **a time of day**.
- **An illumination on a specific date** (a dedicated show): the wait **a date (every year)**, for example “24/12 17:00”.

## Testing without waiting for the evening

- In the editor, select a scene: it is displayed with its final state; ▶ plays its timeline. On reaching a wait, playback stops: another click on ▶ makes it continue without waiting for the deadline.
- For the duration of the tests, replace the wait's rule with a short **length of time** (“30 s”) and watch the whole cycle run in **Show mode**. Remember to put the real rule back afterwards.
- The wait window's preview always shows the next occurrence: it is the surest way to check a rule with excluded days.

## With the standalone player

The standalone player replays the show without a computer, on a small module: **File > Export for a standalone player...**. The export computes the show in advance as a DMX stream; time-of-day waits are **ignored** there, and a looping scene is not repeated endlessly.

For an architectural installation, export only the animation scene (fields **Scenes from** ... **to** ...), and leave the schedule to the player, which has its own weekly programme and a sunset trigger. See The standalone player. The programme is set in the player's `player.ini` file, which its web page lets you edit. Each line of the `[schedule]` section describes a slot: the days (seven digits, Monday to Sunday, 1 = yes), the start, the end, the clip to play, then `loop` (repeating) or `once` (one time). The start and end can be a time, or `sunset` / `sunrise` with an offset in minutes; a slot may run past midnight. Example: every evening, from sunset minus 15 minutes to 23:30, the “soir” clip in a loop:

```
soiree = 1111111 sunset-15 23:30 soir loop
```

Outside the slots, the player blacks out ( `idle = black` ) or keeps the last picture ( `idle = hold` ). The location's latitude and longitude are set in the `[player]` section of the same file.

## Good to know

- After a power cut, DreamShow asks at restart whether to resume at the same scene: someone has to answer. If the cut happened during a wait, the resume replays that wait; if its time went by during the cut, the show carries on directly. See Recovering after a crash or power cut.
- Daylight saving time is handled by the Windows clock: check that the computer has the right time and the right time zone.

- A wait whose rule is impossible (badly written time) is ignored when reached: always check the preview before confirming.
- Parallel sequences freeze during the main show's waits: do not use them to animate a waiting scene.

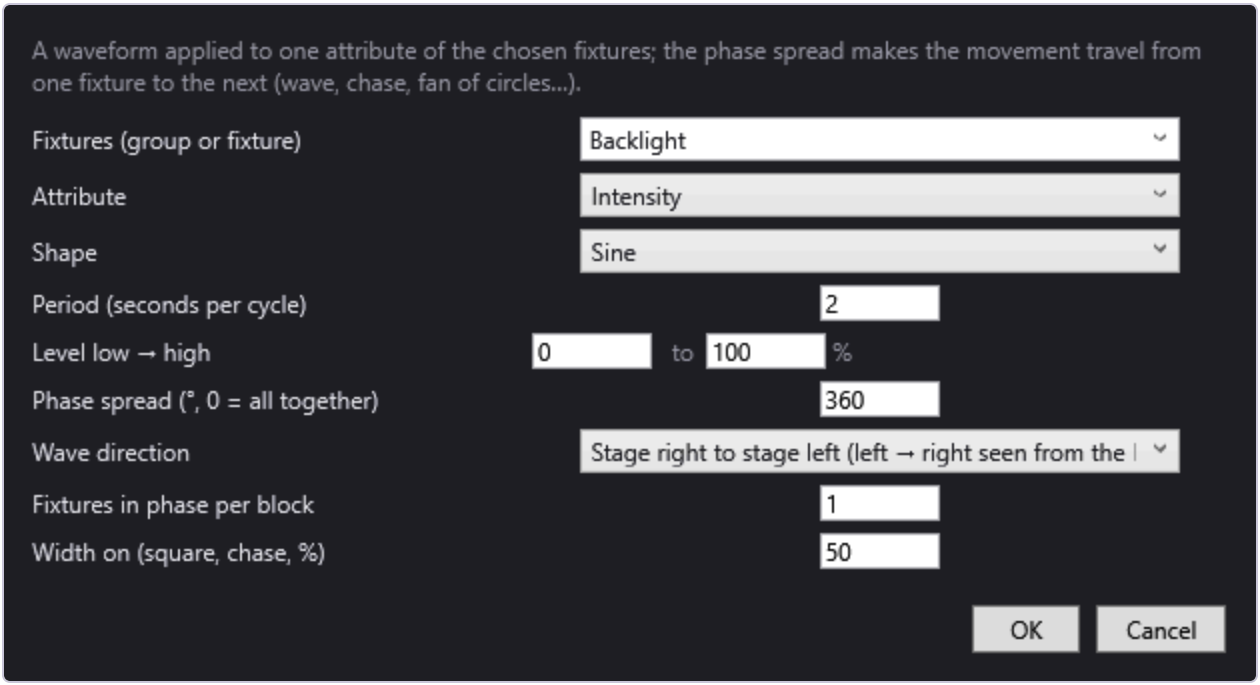
## See also

- Waits (key, time, date, sun)
- Jumps and scene labels
- Scene settings
- Dynamic effects
- Pixel mapping
- The standalone player
- Show mode and the cue window

# Dynamic effects

A dynamic effect applies a waveform (sine, square, chase, random…) to an attribute of a group of fixtures: their intensity, their colour, their position. A phase spread makes the movement travel from one fixture to the next: an intensity wave, a chase, a scrolling rainbow, a fan of moving-head circles. The effect is a timeline block: it is placed, resized and faded like the others.

**To open it:** drag **Dynamic effect…** from the **Others** list (**Libraries** panel) onto a track of the **Scene timeline** panel. The **Dynamic effect** window opens; a double-click on the block reopens it.




## The “Dynamic effect” window

At the top, a reminder: “A waveform applied to one attribute of the chosen fixtures; the phase spread makes the movement travel from one fixture to the next (wave, chase, fan of circles…).”

### Fixtures (group or fixture)

The group or fixture that plays the effect. The list first offers the venue's groups, then each fixture by its identifier; you can also type the name. It is mandatory (“Choose a group or a fixture.”). One group or fixture per effect: for two groups, place two effects.

### Attribute

What the effect varies:

- **Intensity** (default): the fixtures' dimmer.
- **Colour (pulse)**: the colour chosen in the **Colour** line, more or less bright according to the wave.
- **Rainbow**: the hue goes round the colour wheel at each period.
- **Pan**: the moving heads' horizontal position.
- **Tilt**: the moving heads' vertical position.
- **Circle (pan + tilt)**: each moving head describes a circle.
- **Figure eight (pan + tilt)**: each moving head describes a sideways figure eight (∞).

The effect sets **only** the channels of this attribute. The rest (colour, shutter, intensity…) comes from the scene's final state or from other blocks: to see moving heads move with **Pan** or **Circle**, they must be lit by other means.

### Shape

The shape of the wave, over one period (greyed out for **Rainbow**, **Circle** and **Figure eight**, which have their own movement):

- **Sine** (default): soft rise and fall; each cycle starts at the low level.
- **Triangle**: rise then fall in a straight line.
- **Square**: on (high level) for the **width on**, then off (low level).
- **Sawtooth**: steady rise, then a sudden return to the bottom.
- **Falling ramp**: sudden return to the top, then a steady fall.
- **Chase**: like **Square**; with a phase spread and a short width, a single light runs from one fixture to the next.
- **Random**: a level drawn at random at each period, different for each fixture (or each block), held for the whole period.

### Period (seconds per cycle)

Length of one wave cycle, in seconds (2 by default). Minimum 0.05 s (“Invalid period (at least 0.05 s).”). Short period = fast effect.

### Level low → high

The wave's two limits, in % (0 to 100 by default):

- **Intensity**: minimum and maximum intensity;
- **Colour (pulse)**: from the colour off (0 %) to the full colour (100 %);
- **Pan** and **Tilt**: the amplitude of the movement, as a % of the moving head's total range (for example 30 to 70 %);
- **Rainbow**: only the high level counts (the brightness of the colours);
- **Circle** and **Figure eight**: not used (see **Centre and size**).

### Phase spread (°, 0 = all together)

The offset between the fixtures, in degrees of cycle (360 by default):

- **0**: all fixtures do the same thing at the same time;
- **360**: the fixtures are spread evenly over a complete cycle: a wave that runs through the whole group;
- **180**: spread over half a cycle;
- beyond 360: several waves at once over the group.

With a positive spread, the wave runs through the fixtures in the order chosen in **Wave direction** (for example from stage right to stage left with “Stage right to stage left”); with a negative spread (-360), it runs through them the other way.

### Wave direction

The order in which the fixtures receive their offset, from their position in the venue:

- **Stage right to stage left (left → right seen from the hall)** (default);
- **Stage left to stage right**;
- **Downstage to upstage**;
- **From the centre outwards**: those closest to the centre first; symmetrical fixtures receive neighbouring phases, which gives a mirror effect;
- **Patch order**: the order of the fixtures in the group.

The positions are those of **Venue > Fixtures, addresses, positions…**.

### Fixtures in phase per block

Number of neighbouring fixtures (in the chosen order) that share the same phase (1 by default). With 2, the fixtures go in pairs; with 4 on a bar of 12, the wave advances in blocks of 4.

### Width on (square, chase, %)

For **Square** and **Chase**: the part of the cycle where the fixture is at the high level (50 % by default, minimum 2 %). For a chase with a single fixture lit out of N, set about 100/N %.

### Colour

Visible only with **Colour (pulse)**: the colour, as “red,green,blue 0-255” (255,255,255 by default, for example “255,80,0” for an orange).

### Centre and size

Visible only with **Circle** and **Figure eight**:

- **pan** and **tilt**: the centre of the movement, as a % of the range (50 and 50 by default: the middle);
- **radius**: the size of the movement, as a % of the range (10 by default, at most 50).

For the **Figure eight**, the radius is the amplitude in pan; the amplitude in tilt is half of it.

### OK / Cancel

**OK** checks the values and creates (or updates) the block. **Cancel** on creation: no block is created.

## The block on the timeline

- Deep purple colour, text "∿ Intensity sine · Backlights · 2 s · 360°" (attribute, shape, target, period, spread).
- Initial length: 10 s. Resize it with the handle on the right edge: the wave continues for the whole length (20 minutes at most).
- **Fades**: right-click > **Properties…** gives **Start (s)**, **Length (s)**, **Fade in (s)** and **Fade out (s)**. As for a light block, the fade only concerns intensity; a position (Pan, Circle…) applies from the start of the block.
- The effect is computed in advance, at 25 frames per second: it follows the red line, replays identically and is displayed in the simulation.
- It is laid **on top of** the light blocks and the final state, on the channels it sets (see Light and effect blocks). A **Blackout** switches it off.
- During a wait block, the effect stops on the spot.

## Step by step

### A soft wave on an LED cyclorama

1. In the scene's final state, light the "Cyclo" group in blue at 100 %.
2. Drag **Dynamic effect…** onto a track at 0 s.
3. **Fixtures**: "Cyclo"; **Attribute**: **Intensity**; **Shape**: **Sine**; **Period**: 4; **Level low → high**: 20 to 100; **Phase spread**: 360; **Wave direction**: **Stage right to stage left**. **OK**.
4. Resize the block to the length of the scene; right-click > **Properties…**, **Fade in (s)** 3.

If the cyclorama's fixtures have no dimmer channel (only red, green, blue), **Intensity** has no effect: use **Colour (pulse)** with the colour "0,0,255".

### A chase on a bar of 8 fixtures

1. **Attribute**: **Intensity**; **Shape**: **Chase**; **Period**: 1; **Level** 0 to 100; **Spread**: 360; **Width on**: 12 (about 100/8).
2. Light the bar in the final state (colour), with the effect doing the intensity.

### A fan of circles with the moving heads

1. In the final state, light the moving heads (intensity, colour, shutter open) and aim at the centre of the stage.
2. **Dynamic effect…**: **Attribute**: **Circle (pan + tilt)**; **Period**: 6; **Spread**: 90; **Centre and size**: pan 50, tilt 40, radius 8.
3. The moving heads describe circles offset by a quarter turn from one another.

### Random flashes

1. **Attribute**: **Intensity**; **Shape**: **Random**; **Period**: 0.3; **Level** 0 to 100; **Spread**: 0; **Fixtures in phase per block**: 1.
2. Each fixture takes a random level every 0.3 s.

## Good to know

- A dynamic effect targets only one group or fixture. To chain two effects, place two blocks; two overlapping blocks do not mix with each other (mixing only concerns the light and effect blocks from the library).
- **Square** and **Chase** give the same result; the name only helps you remember the intention.
- With **Colour (pulse)** on a moving head with a colour wheel, DreamShow chooses the nearest wheel colour: the pulse may make the wheel change position.
- The effects in the **Effects** list (library) are fixed settings; dynamic effects are movements. They are not stored in the library: copy the track (right-click on its name > **Copy the track**) to reuse them in another scene.

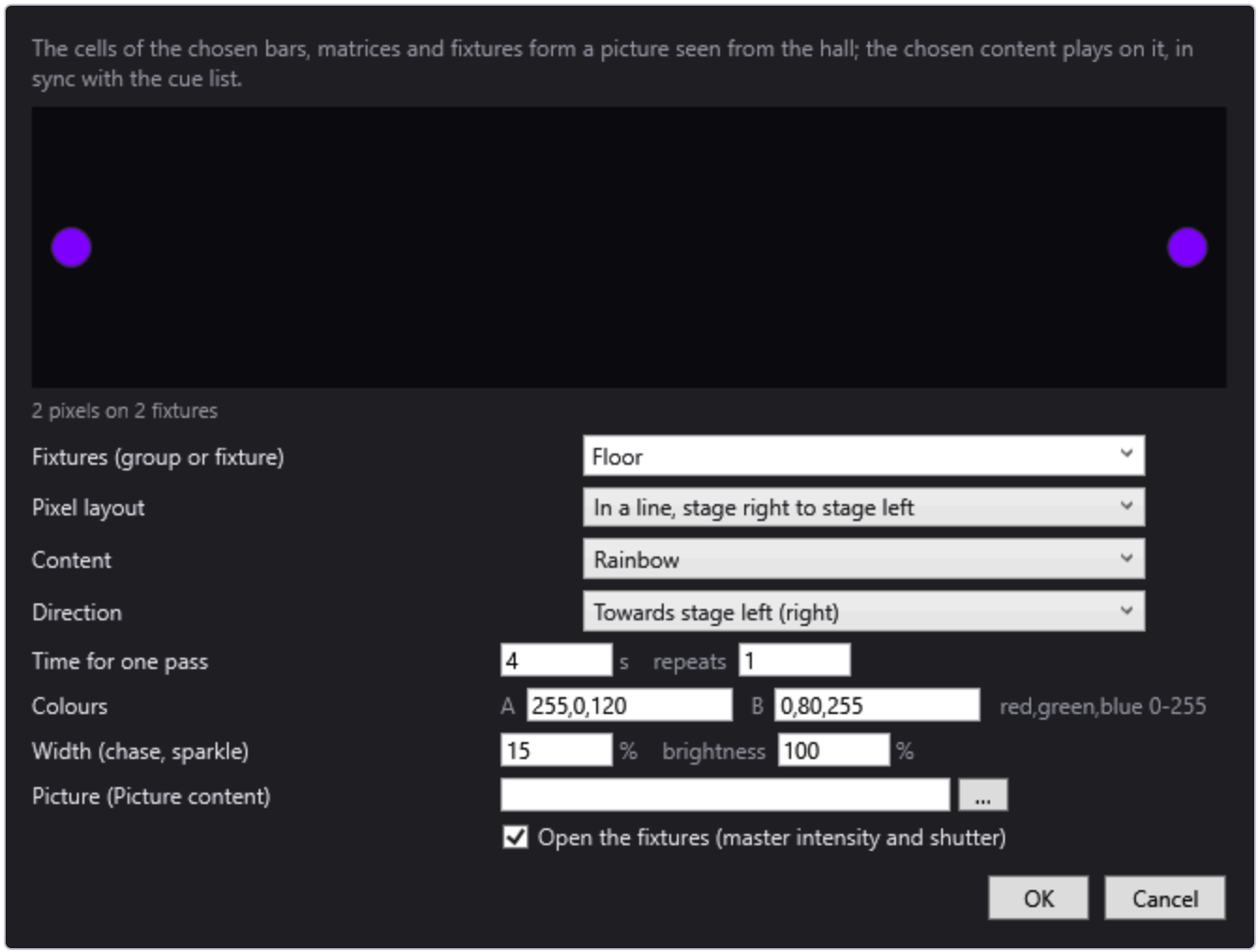
## See also

- The scene timeline
- Light and effect blocks
- Fades and mixes
- Pixel mapping
- Trajectories
- Fixtures, addresses, positions

# Pixel mapping

Pixel mapping treats the cells of your LED bars, matrices and LED fixtures as the pixels of a screen, seen from the hall, and plays an animated picture on it: scrolling gradient, rainbow, wave, plasma, chase, sparkle, fire, or a picture of your choice. It is a timeline block, synchronised with the cue list.

**To open it:** drag **Pixel mapping…** from the **Others** list (**Libraries** panel) onto a track of the **Scene timeline** panel. The **Pixel mapping** window opens; double-clicking the block reopens it.




## What a pixel is here

Each **coloured cell** of a fixture becomes a pixel: an LED bar with 8 cells gives 8 pixels, a 5 × 5 matrix gives 25, an ordinary LED fixture (a single colour for the whole fixture) gives 1. DreamShow finds the cells from the fixture definition (the red, green, blue and white channels of each cell). See The fixture library.

## The “Pixel mapping” window

At the top, a reminder: “The cells of the chosen bars, matrices and fixtures form a picture seen from the hall; the chosen content plays on it, in sync with the cue list.”

### The preview

Under the reminder, a black frame shows each pixel as a dot, in its place, and plays the content live: every setting you change is visible at once. Below it, a counter: “48 pixels on 6 fixtures”, or “No colour cell in this target.” if the target contains no usable fixture.

### Fixtures (group or fixture)

The group or fixture that receives the picture (list of groups, then of fixtures; you can type the name). Required (“Choose a group or a fixture.”).

### Pixel layout

How the pixels are arranged in the picture:

- **In a line, stage right to stage left** (default): all the pixels on a single line, fixture after fixture, from left to right as seen from the hall (order of the positions in the venue), cell after cell.
- **One row per fixture (matrices unfolded)**: each fixture forms a row, the highest at the top; a square matrix (4 cells or more: 2 × 2, 5 × 5, 8 × 8…) is unfolded into a square.
- **By where the fixtures hang (seen from the hall)**: each pixel in its real place, seen from the front: from left to right according to the position in the venue, from top to bottom according to the height. The cells of a bar are spread along the bar (10 cm between two cells, taking its orientation into account).

Positions and heights come from **Venue > Fixtures, addresses, positions…**.

### Content

The picture played on the pixels:

- **Scrolling gradient A → B**: a soft gradient from colour A to colour B and back, moving along.
- **Rainbow** (default): all the hues, scrolling. Colours A and B are not used.
- **Wave of colour A**: colour A fading in and out in a wave, with black between two waves.
- **Plasma A / B**: a moving, undulating blend of colours A and B (the **Direction** has no effect here).
- **Chase of colour A**: a band of colour A running on a black background; its width is set by **Width**.
- **Sparkle A over B**: pixels that light up at random in colour A over a background of colour B; the proportion of lit pixels is set by **Width**, the rhythm by **Time for one pass** (a new draw

every twentieth of a pass).
- **Fire**: orange flames, hotter at the bottom, flickering (about 12 times per second). The colours, the direction and the time for one pass are not used.
- **Image**: the picture chosen in **Picture (Picture content)**, scrolling in the chosen direction.

### Direction

The direction in which the content moves:

- **Towards stage left (right)** (default);
- **Towards stage right (left)**;
- **Upwards**;
- **Downwards**.

**Upwards** and **Downwards** only make sense with a layout on several rows (**One row per fixture** or **By where the fixtures hang**): in a line, all the pixels would take the same colour.

### Time for one pass

- **s**: the time, in seconds, for the content to cross the picture once (4 by default; minimum 0.05). Smaller = faster.
- **repeats**: how many times the pattern repeats across the width of the picture (1 by default). With 2, you see two rainbows side by side. For **Plasma**, this setting changes the fineness of the undulations.

### Colours

**A** and **B**, each as “red,green,blue 0-255” (by default A = 255,0,120, a pink, and B = 0,80,255, a blue). A badly written value is ignored (the previous one is kept).

### Width (chase, sparkle)

- **%**: for **Chase**, the width of the band (in % of the picture, 15 by default); for **Sparkle**, the proportion of pixels lit in A.
- **brightness**: the overall intensity of the content, in % (100 by default).

### Picture (Picture content)

The picture file to play (.png, .jpg, .jpeg, .bmp, .gif). The **...** button opens the file chooser (in the show's media folder) and automatically switches the **Content** to **Image**. A picture from the media folder is saved by its name only (the show stays movable together with its folder); a picture taken from elsewhere is saved with its full path. The picture is reduced to 256 pixels wide at most. With the **Image** content, a file is required (“Choose the picture to play.”).

The picture is read as a strip: with **Towards stage left** or **Towards stage right**, it scrolls horizontally across the full width of the pixels; with **Upwards** or **Downwards**, vertically.

### Open the fixtures (master intensity and shutter)

Ticked (default): the block also sets the fixtures' master dimmer to 100 % and opens their shutter, so that the picture is visible even if nothing else lights them. Unticked: only the cell colours are set; the master intensity must come from the final state or from another block.

### OK / Cancel

**OK** checks the settings and creates (or updates) the block. **Cancel** at creation: no block is created.

## The block on the timeline

- Teal colour, text "⣿ Rainbow · Bars · 4 s" (content, target, time for one pass).
- Starting length: 10 s; resizable (20 minutes at most).
- Right-click > **Properties…**: **Start (s)**, **Length (s)**, **Fade in (s)**, **Fade out (s)**. The fades bring the picture up and down.
- The picture is calculated in advance, at 25 frames per second: it follows the red line and is shown in the simulation. If you replace the picture file, DreamShow takes it into account at the next recalculation of the show (at the next change to the timeline or the settings).
- The block lies on top of the light blocks and the final state, on the channels it sets. A **Blackout** turns it off. During a wait block, the picture freezes.

## Step by step

### A rainbow running across the LED bars at the back

1. Create a group "Bars" with your LED bars (see Libraries).
2. Drag **Pixel mapping…** onto a track at 0 s.
3. **Fixtures**: "Bars"; **Layout**: **In a line, stage right to stage left**; **Content**: **Rainbow**; **Direction**: **Towards stage left (right)**; **Time for one pass**: 6 s, **repeats** 1. Watch the preview, then **OK**.
4. Stretch the block over the length of the piece; **Properties…**: **Fade in (s)** 2, **Fade out (s)** 4.

### A logo scrolling across a matrix

1. Put "logo.png" in the show's media folder.
2. **Pixel mapping…**: **Fixtures**: the matrix; **Layout**: **One row per fixture (matrices unfolded)**; **…** to choose "logo.png" (the content switches to **Image**); **Direction**: **Towards stage right (left)**; **Time for one pass**: 8.

### A fireplace fire on a wall of LED PARs

1. Place the PARs at their real height in the venue.
2. **Layout**: **By where the fixtures hang (seen from the hall)**; **Content**: **Fire**; **brightness** 80.
3. The fixtures at the bottom burn brighter than those at the top, and flicker.

## Good to know

- A fixture without a colour channel (dimmer only) counts as one pixel but shows nothing: pixel mapping sets colours. With a white channel only, the pixel receives the average of the three colours.
- On a red, green, blue and white cell, the white receives the smallest of the three colours.
- The preview in the window runs continuously; on stage, the picture starts at the beginning of the block.
- The pixels of the **In a line** layout are ordered according to the left-right position of the fixtures: set the positions in the venue so that the order is right.

## See also

- The scene timeline
- Dynamic effects
- Light and effect blocks
- The fixture library
- Fixtures, addresses, positions
- 2D and 3D simulation

# Aiming and covering a zone with moving heads

Two tools in the **Venue** panel set the pan and tilt of moving heads without touching the sliders: **Aim** points all the selected moving heads at the same spot, **Zone** spreads the selected moving heads over a surface that you draw (rectangle or ellipse), on the floor, on the stage or on a wall. DreamShow calculates the angles from each moving head's position, its hanging and its pan / tilt travel, or from its calibration if it has one.

**To get to it:** the **Aim** and **Zone** buttons in the toolbar of the **Venue** panel (the simulation), on the main screen.




## What these tools modify

Aim and Zone write the pan and tilt into **what you are currently editing**:

- normally, the **final state of the scene** selected (the show is then marked as modified);
- when editing an effect, that effect;
- in trajectory mode, the selected **key frame** (see Trajectories).

For each moving head aimed, DreamShow also adds, if it does not already have them in that state: intensity at 100 %, shutter open and the fastest pan / tilt speed. A position coming from a palette is detached from the palette (see Palettes).

If **LIVE** is on and the scene being edited is the one playing, the real moving heads move at once.

Only **moving heads** are concerned: fixtures whose definition has a **Max pan (°)** greater than 0, and which are placed in the venue. The other fixtures selected are ignored.

## Selecting the moving heads

Before aiming, select the moving heads to point. Several ways:

- the **Selection — groups and fixtures** tree in the **Scene settings** panel: tick fixtures or whole groups;
- a click on a fixture's dot in the **Venue** panel ticks or unticks that fixture;
- in 2D, a drag on an empty area (**Aim** and **Zone** tools off) draws a selection rectangle; hold **Ctrl** to add to the selection instead of replacing it;

- in 3D, a click (without dragging) on a fixture ticks or unticks it;
- **All** selects all the groups and the fixtures outside groups, **None** empties the selection.

See also Selecting fixtures and adjusting the light.

## Aim

- **Aim**: toggle button. When on, a click in the venue points **all** the selected moving heads at the point clicked: their beams converge. Hold the mouse button down and move it: the point follows the mouse.
- In 2D, the point aimed at is on the floor, under the mouse.
- In 3D, the point aimed at is the surface under the mouse: the stage, the venue floor or a wall. In 3D, the left click aims; the right-click drag always rotates the view, which lets you aim and orbit around the venue without changing tool.
- If no moving head is selected, the information line at the top of the main screen displays “Aim: select one or more moving heads first”.
- Turning **Aim** on turns **Zone** off.

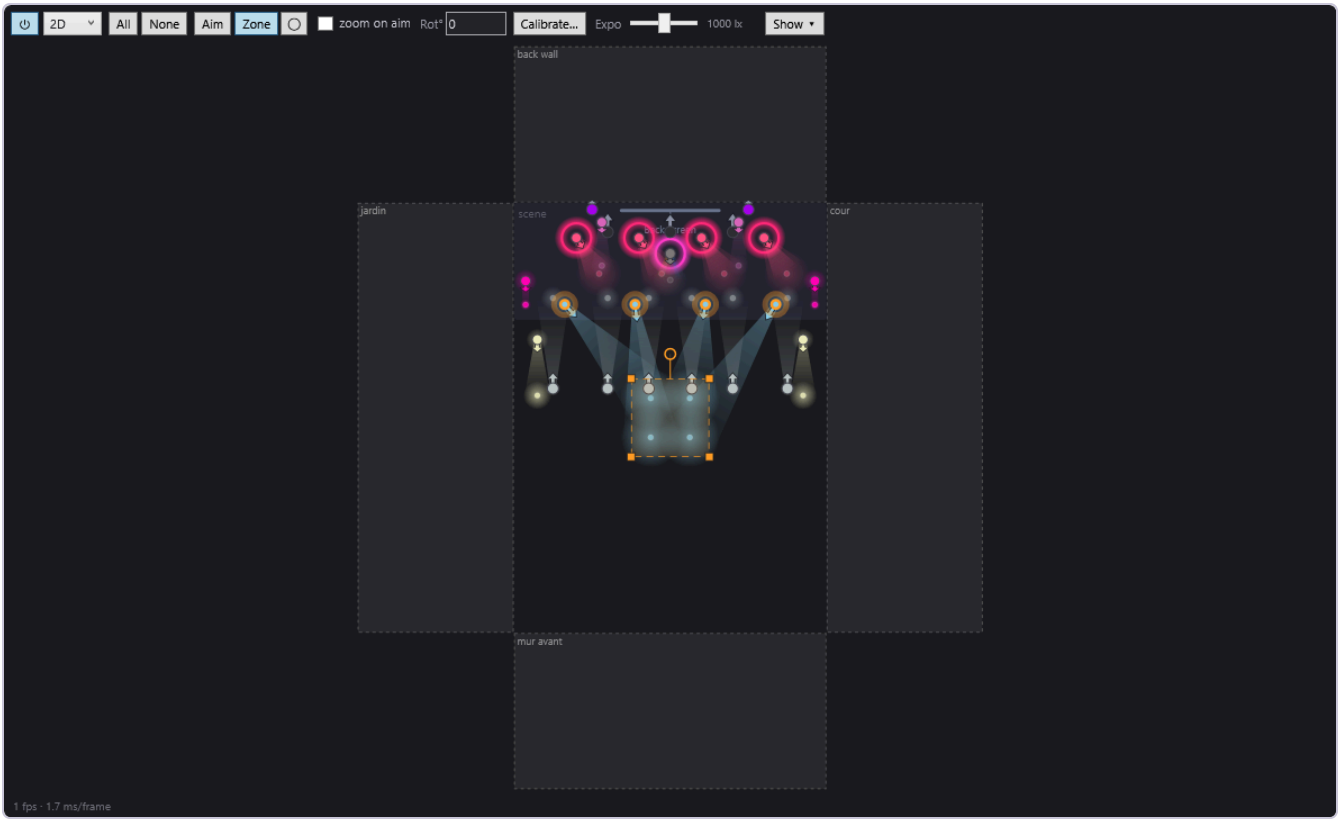
**zoom on aim** (check box, unticked by default, remembered from one session to the next):

- unticked, **Aim** only sets pan and tilt: your zoom is kept;
- ticked, aiming also tightens the beam to make a spot about 1 m wide on the target. This only works for a moving head whose definition declares its beam angles (**Beam (°) from … to …**) and whose mode has a zoom (or focus) channel.

When a moving head has a calibration and the point aimed at is on the stage, DreamShow uses the calibration, which is more exact than the geometric calculation. See Moving-head calibration.

## Zone

**Zone**: toggle button. When on, you draw a zone on the venue and the selected moving heads spread themselves over it.




### The plan with unfolded walls

In Zone mode, the 2D plan gets larger: around the venue floor, the four walls are “unfolded” flat, like an open box (each strip has the height of the venue). A zone drawn in the back-wall strip is on that wall; a zone pushed from the floor against a wall **climbs it gradually** instead of jumping. Along the stage, the climb starts at stage-floor level.

In 3D, the zone is drawn directly on the surface under the mouse (stage, floor or wall).

### Drawing and modifying the zone

- **Draw**: press and drag on an empty spot: the zone is the rectangle (or ellipse) between the starting point and the end point. A simple click without dragging creates nothing.
- **Move**: drag **inside** the zone. The moving heads follow live.
- **Resize**: drag one of the four **corner handles** (small orange squares). The zone grows or shrinks symmetrically around its centre; its centre and its angle do not move.
- **Rotate**: drag the **circle** placed above the zone. Turns add up: turning twice makes 720°.
- ◯: the zone's shape. Pressed = ellipse, released = rectangle (default). Changing the shape also applies to the zone already drawn.
- **Rot°**: the zone's angle in degrees, typed on the keyboard. Values beyond one turn are allowed (360 = 1 turn, 720 = 2 turns, 1080 = 3 turns). Confirm with **Enter** or by leaving the field. The field shows the current angle when you rotate with the mouse. In a trajectory, between two key frames, the angle is interpolated as it is: from 0 to 720, the zone makes two turns.

The zone stays displayed as long as **Zone** is on: it is the handle that drives the moving heads. Turning **Zone** off erases the zone; the moving heads keep their aim.

### How the moving heads are distributed

- The zone is cut into a grid of rows that are as square as possible, and the moving heads are shared equally between the rows (for example 13 moving heads on 3 rows: 5 + 4 + 4, the fullest row in the middle).
- The moving heads are assigned to the cells in the order of their position (from stage right to stage left, then front to back), so that the beams do not cross. This assignment stays the same when the zone rotates or moves: each moving head keeps its place in the zone.
- If the moving head's definition declares its beam angles and its mode has a zoom (or focus) channel, the beam opening is set to cover its cell (the cell's diagonal). Zone mode always sets the zoom this way: that is its job, unlike **Aim**.
- Moving heads marked **Out of order** in the venue are ignored.
- The information line at the top of the main screen summarises the result, for example “Zone (floor, rectangle): 6 head(s) over 4.0 × 2.0 m (3×2)” (surfaces touched, shape, number of moving heads, size, cells). With no moving head selected: “Zone: select one or more moving heads first”.

## Other buttons in the bar

- **Calibrate…**: 4-point calibration of the selected moving head. See Moving-head calibration.
- The bar's other settings (⏻, 2D / 3D, quality, **Expo**, **Show** ▾) are described in 2D and 3D simulation.

## Step by step

### Making 4 moving heads converge on an actor at the centre of the stage

1. Select the scene in the **Scenes** panel.
2. Select the 4 moving heads (tick the “Moving heads” group, for example).
3. Set their intensity and colour in **Scene settings** if needed.
4. Turn **Aim** on.
5. Click at the centre of the stage (avoid clicking right on a fixture's dot: that click ticks or unticks the fixture). Hold and adjust by moving the mouse.
6. For a tight spot, tick **zoom on aim** before clicking.
7. Turn **Aim** off and save the show.

### Covering the whole downstage area with a carpet of light

1. Select the scene, then the moving heads to use.

2. Turn **Zone** on (the plan gets larger with the walls unfolded).
3. Draw a rectangle on the downstage area, from one edge to the other.
4. Adjust with the corner handles until the spots touch without overlapping too much.
5. Turn **Zone** off and save.

### Dressing the back wall

1. Select the moving heads, turn **Zone** on.
2. Draw the zone in the back-wall strip (above the floor on the 2D plan), or switch to 3D and draw directly on the wall.
3. To make it rotate slowly during the scene, create a trajectory with two zone key frames, one at **Rot°** 0 and the other at 360 (see Trajectories).

## Good to know

- **Exact positions are essential**: aiming is calculated from the moving heads' X, Y, Z positions, their orientation, **Head up (standing on the floor)** and the **Max pan (°)** / **Max tilt (°)** travel of the definition. A moving head badly placed in the venue will aim beside the target. For centimetre accuracy on the stage, calibrate.
- A click on a fixture's dot selects that fixture instead of aiming: start your gestures in an empty area.
- **Aim** and **Zone** are never on together.
- What you aim at is saved in the scene (or effect, or key frame) being edited: check in **Scenes** that it is the right one before aiming.
- The zoom is only set if the definition declares its beam angles and has a zoom or focus channel.
- A calibration is used by **Aim** and **Zone** for any point located on the stage, even raised.

## See also

- Trajectories
- Moving-head calibration
- Fixtures, addresses, positions
- The fixture library
- Selecting fixtures and adjusting the light
- 2D and 3D simulation
- The follow spot on a tablet

# Trajectories

A trajectory animates moving heads (or other fixtures) over time: you place **key frames**, each with a state of the fixtures (position, colour, intensity, zoom…), and DreamShow calculates everything that happens between two keys. Placed on the timeline as a block, the trajectory makes the stage sweep, a carpet of light rotate or a colour glide during a scene, for example.

The key frames are set with the usual tools of the main screen (**Aim**, **Zone**, sliders, palettes, calibration): while editing, these tools write into the selected key.

**To get to it: Libraries** panel, **Trajectories** section: **+ trajectory (selection)** button, list of trajectories (double-click to open the editor). On the timeline, double-clicking a trajectory block also opens its editor.




## The trajectory library

Trajectories are separate files (the `trajectoires` folder of the data, one `name.traj.json` file per trajectory). They are shared by all shows: the same trajectory can be placed in several scenes and several shows.

In the **Libraries** panel, **Trajectories** section:

- **The list**: the trajectories in alphabetical order. Double-click = open the editor; drag a trajectory onto a timeline track = insert it there; **Del** key = delete.
- **Right-click** on the list:
  - **Open the editor**;
  - **Duplicate…**: copies the trajectory under a new name (“Name of the copy of “…”:”, suggested “name (copy)”). The original and the blocks that use it do not change: this is the safe way to make a variant;
  - **Delete…**: after confirmation, the file is erased for good. Blocks already placed on a timeline stay, but no longer play anything.
- **+ trajectory (selection)**: creates a trajectory for the ticked fixtures or groups. Their current state becomes the first key frame (at 0 s), then the editor opens.

## Creating a trajectory

1. Tick the moving heads (or groups) that are going to move, and place them where the trajectory should start (Aim, Zone, sliders…).
2. Click **+ trajectory (selection)**. With no selection, DreamShow reminds you: “Tick the moving heads (or groups) of the trajectory first and place them: their state becomes the first key frame.”
3. Type the **Trajectory name:**. A name already taken is refused (“A trajectory already has that name.”).
4. The editor opens in **trajectory mode**.

The fixtures targeted (the “targets”) are fixed at creation; they are listed at the top of the editor.

## Trajectory mode

As long as the editor is open, the main screen is in trajectory mode:

- the banner of the **Scene settings** panel is orange: “TRAJECTORY MODE — “name”: the settings change the selected key frame”;
- the final state list becomes “Key frame of “name””;
- the selection is replaced by the trajectory's targets;
- **Aim**, **Zone**, the sliders, the palettes and the calibration write into the selected key frame, and the simulation shows it.

When a key is selected, the banner says “… — key at t = 2.00 s” (followed by “(zone)” if it is a zone key). During a preview between two keys, it says “… — preview at t = … s (not a key)”: what you set at that moment **is not kept**. Select a key (or create one) before adjusting.

The mode ends when you close the editor (**Close** button, or the banner's **✓ Finish editing the effect** button, which closes the editor in this mode).

## The Trajectory window

The title is “Trajectory — name”. At the top, a reminder of the targets and gestures.

### Playback

- ⏮ **Start**: brings the preview marker back to 0 s.
- ▶ **Play** / ⏸ **Pause**: plays the interpolated trajectory in the simulation: the moving heads move and change colour. Playback stops at the end, on the last pose. At least two keys at different times are needed (“Add at least two keys at different times.”).

### The keys

- ◀ **Key** / **Key** ▶: selects the previous / next key.
- **+ Key at the preview**: new key frame at the time of the preview marker, with the state interpolated at that moment (and the interpolated zone, on a section driven by a zone).
- **+ Key at the end (+2 s)**: new key 2 s after the last one, a copy of the state at the end.
- **Duplicate**: copies the selected key 1 s further on.
- **− Delete**: deletes the selected key. A trajectory keeps at least one key.

A key is never placed exactly on another: if the time is already taken, it is shifted to just after.

### The properties of the selected key

- **Key — time (s):** the key's time, in seconds from the start of the trajectory. Confirm with **Enter** or by leaving the field; the keys reorder themselves.
- **To the next key:** the way of going from this key to the next:
  - **linear** (default): at constant speed;
  - **eased**: starts and arrives gently;
  - **hold**: stays on this key until the next one, then jumps. On the strip, the segment is dotted.
- On the right, “preview t = … s / … s”: the marker's time and the total length.

### The key strip

- A seconds ruler at the top; each key is a numbered **diamond**; the selected key is orange; the preview marker is the red line.
- **Clicking a diamond**: selects the key. Its settings then follow the main window's tools.
- **Dragging a diamond**: moves the key in time (to the hundredth of a second). The simulation follows.
- **Clicking or dragging on an empty area**: moves the preview marker; the simulation shows the interpolated state.
- **Double-clicking on an empty area**: new key at that spot.

### The bottom of the window

- **Save**: saves the trajectory. If it is already placed on timelines, DreamShow asks what to do: **Yes** replaces the trajectory everywhere (all those blocks change), **No** saves it under a **new** name (the original stays intact, and the editor carries on with the copy), **Cancel** saves nothing.
- **Track:** the track of the selected scene where the block is to be inserted.
- **Insert into the timeline**: saves the trajectory if needed, then adds a trajectory block on the chosen track, at the red line's time if the selected scene is the one playing (otherwise at 0 s). The block lasts the length of the trajectory (rounded up to the next tenth of a second, 0.5 s at least). A scene must be selected.
- **Close**: closes the editor and leaves trajectory mode. If changes remain: “Save the trajectory before closing?” (**Yes**, **No**, **Cancel**).
- A help line confirms the actions (“Trajectory saved.”, “Block inserted into the timeline.”, “Key N selected: …”).

## What is interpolated

Between two neighbouring keys, DreamShow blends each setting of each target: intensity, pan, tilt, colour, and the channels set by hand (zoom, cells…).

- A setting present in only one of the two keys keeps its value over the whole section. Exception: the colour of a cell (Red 1, Green 2…) set on a single key fades from the other key's general colour.
- Before the first key, the first key applies; after the last, the last.

### Zone keys

If you use the **Zone** tool on a key, the zone is saved in the key: it is a **zone key**. Between two zone keys, it is the zone itself that is interpolated (centre, size, rotation) and the moving heads are distributed over it again at each instant: a rotation makes the spots turn around the zone's centre, instead of each moving head taking the shortest path. Colour and intensity keep interpolating normally. A section in which only one key has a zone keeps that zone from one end to the other.

A zone's angle is interpolated as it is: from **Rot°** 0 to **Rot°** 720, the zone makes two complete turns. For zone movements, DreamShow chooses for each moving head the pan direction that stays in the middle of its travel throughout the movement, to avoid it hitting its end stop and going back a full turn.

During preview and playback, the interpolated zone is drawn on the plan and in the 3D view.

## The trajectory block on the timeline

- **Insertion**: **Insert into the timeline** in the editor, or drag the trajectory from the **Trajectories** list onto a track (the block takes the length of the trajectory, or 10 s if it has no length).
- **Length**: the whole trajectory is stretched over the block's length. A block longer than the trajectory plays it more slowly, a shorter block faster.
- **End**: if the block is the last to finish on its track, the fixtures keep the last pose after the end of the block.
- **Fades**: the block has, like the others, a fade in and a fade out (see Fades and mixes).
- **Double-click** on the block, or right-click > **Edit the trajectory…**: opens the editor on that trajectory. If the file no longer exists: "Trajectory "…" not found or unreadable."
- Modifying then saving a trajectory updates all the blocks that use it (unless you choose to save it under another name).

See Light and effect blocks and The scene timeline.

## Step by step

### A left-to-right sweep in 6 seconds

1. Select the scene, tick the moving heads.
2. With **Aim**, point them at the stage-right side of the stage.
3. **+ trajectory (selection)**, name it "Sweep".
4. In the editor, **+ Key at the end (+2 s)**: a key 2 appears at 2 s. Type **6** in **Key — time (s):** then **Enter**.
5. With key 2 selected, aim at the stage-left side with **Aim**.
6. Select key 1 and choose **eased** in **To the next key:**.
7. ▶ **Play** to check in the simulation.
8. Choose the **Track:**, place the red line, **Insert into the timeline**, then **Close**.

### A rotating carpet of light

1. Tick the moving heads, turn on **Zone** and draw a rectangle on the stage.
2. **+ trajectory (selection)**, name it "Rotating carpet".
3. Select key 1, then touch the zone (move it slightly, or confirm **Rot°** at 0): the key becomes a zone key.
4. **+ Key at the end (+2 s)**, set its time (for example 20 s), then type **360** in **Rot°** and **Enter**.
5. ▶ **Play**: the spots turn around the zone's centre.
6. Insert the block into the timeline.

### Changing colour at an exact moment with no fade

1. Create two keys at the same position, one red, the other blue.
2. On the first, choose **hold**: the red holds until the time of the second key, then the blue arrives all at once.

## Good to know

- Always adjust **a selected key**: settings made during a preview between two keys are lost.
- A trajectory is a file shared by all shows. To modify a trajectory already in use without changing the other scenes, **Duplicate…** first, or answer **No** to the save question.
- Deleting a trajectory leaves its blocks on the timelines, but they no longer play anything: delete them too.
- The block's length controls the speed: lengthening the block slows the whole movement.
- A trajectory only plays on the targets chosen when it was created.
- Selecting a key counts as a modification: on closing, DreamShow almost always offers to save. Answer **No** if you have changed nothing.

## See also

- Aiming and covering a zone with moving heads
- Moving-head calibration
- The scene timeline
- Light and effect blocks
- Dynamic effects
- Libraries

# Moving-head calibration

To aim, DreamShow computes a moving head's pan and tilt from its position in the venue, its rigging and the pan / tilt ranges of its definition: this is the **geometric model**. It is good, but a head rigged slightly crooked, a bar not quite at the measured height or an approximate definition will make it aim off target. The **4-point calibration** corrects this: for each corner of the stage, you show DreamShow which pan and tilt values actually send the beam there. Afterwards, any point of the stage is aimed at by interpolation between these measurements, as accurately as possible.

**To open it:** select **one** moving head only, then click the **Calibrate…** button in the toolbar of the **Venue** panel. If no moving head is selected, the information line at the top of the main screen shows “Calibrate: select a moving head”; if there are several, “Calibrate: one head at a time”. The head must have a position in the venue and a definition with a pan.

## When to calibrate

- When aiming must be precise on the stage: convergences, follow spot on a tablet, tight zones.
- When a moving head visibly aims off target although its position is right.
- After the moving heads have been re-rigged.

Calibration is done on the real equipment: the DMX output must work (**LIVE** on) and you must be able to see the stage.

## The six points

The “Calibration — (name of the head)” window stays above the other windows and does not block the main screen. It offers six points:

1. **Stage-right corner – upstage**
2. **Stage-left corner – upstage**
3. **Stage-right corner – downstage**
4. **Stage-left corner – downstage**
5. **Back wall – left (top)** (optional)
6. **Back wall – right (top)** (optional)

The four corners are those of the stage declared in the venue (width, offset, depth), on top of the floor. The two back-wall points are halfway up between the stage and the ceiling, directly above the stage-right and stage-left edges of the stage. The four corners are enough; the two wall points extend aiming onto the back wall.

For each point, the line shows on the right the captured value (“pan 48.2 tilt 31.5”, as a percentage of the range) in green, or “—” if there is nothing, and three buttons:

- **Model**: places the head on this point according to the geometric model. The real head moves there: this is the starting point, and only the difference remains to be corrected.
- **Capture**: keeps the head's current pan and tilt for this point. If the head has no pan / tilt set yet: “No current pan/tilt value for this head: set it first (sliders or “Model”).”
- ×: forgets this point.

Under the list, a help line tells you where you stand: “N/4 stage corners captured — all four are needed to save.”, then “The four corners are there: you can save” (“(with the back-wall extension).” if the two wall points are captured too).

Buttons at the bottom:

- **Pre-fill everything (model)**: fills the six points from the geometric model (the head's position, pan / tilt ranges, mounting), without moving the head. Saved as it is, it is a “default” calibration.
- **Save the calibration**: saves the calibration in the venue (immediately, in its file) and closes the window. All four corners are needed (“Capture the four stage corners first.”). The wall points are kept only if both are captured. The log notes “(name): calibration saved” (“(with the back wall)” where applicable).
- **Delete (geometric model)**: removes the head's calibration; aiming goes back to the geometric model. The log notes “calibration removed (geometric model)”.
- **Close**: closes without saving.

If the head already has a calibration, its points are displayed when the window opens.

## How the head is driven during calibration

**Model** and the **Pan** and **Tilt** sliders of the **Scene settings** panel write into the final state of the selected scene (or into the effect or keyframe being edited), like any setting. The head also receives, if it does not have them, intensity at 100 %, the shutter open and the maximum pan / tilt speed. If **LIVE** is on and this scene is the one playing, the real head moves.

So calibrate on a working scene (or a scene you can clean up afterwards), then remove the head from the final state if it must not appear there (**Remove from the final state** button of the **Scene settings** panel).

## Step by step: calibrating a moving head

1. Check the head's position and the stage dimensions in **Venue > Fixtures, addresses, positions...**.

2. Turn **LIVE** on, select a working scene in the **Scenes** panel. Black out the other fixtures to see the beam clearly.
3. Select the head alone, then **Calibrate…**.
4. On the **1. Stage-right corner – upstage** line, click **Model**: the head moves near the corner.
5. Correct with the **Pan** and **Tilt** sliders until the centre of the beam touches the corner exactly (a mark on the floor helps).
6. Click **Capture**: the value appears in green.
7. Repeat for the other three corners.
8. Optional: do the same for the two back-wall points.
9. **Save the calibration**.
10. Check with **Aim**: click at different places on the stage, the beam should land under the mouse.
11. Do the same for each moving head.

## What calibration is used for afterwards

A calibrated moving head uses its calibration, instead of the geometric model, for:

- **Aim** at a point of the stage (see Aiming and covering a zone with moving heads);
- follow spot on a tablet when the finger is on the floor (see Follow spot on a tablet);
- the aims written automatically by the first-show assistant;
- the simulation: the beam's impact point drawn on the stage follows the calibration.

The **Zone** tool uses it in the same way, for any point of the zone located on the stage.

Outside the stage (house, side walls), and above it if the wall points are not captured, the geometric model does the aiming.

## Saving and reloading calibrations

Calibrations are part of the venue: they follow **Duplicate the venue…** and the project export. To save them separately or to take them over from a ppshow installation: **Venue > Export the calibrations…** and **Venue > Import the calibrations (lyrescal.ini)…** (see Switching, renaming, duplicating a venue).

## Good to know

- A calibration is valid for **one** rigging position: if the head is moved or re-rigged, recalibrate it.
- **Duplicating** a fixture in the venue does not copy its calibration.
- Changing a fixture's identifier does not make it lose its calibration (it is saved with the fixture), but a calibration file exported earlier will no longer find it under its old identifier.
- If the **Fixtures, addresses, positions…** window is open during calibration, saving the venue from it afterwards would erase the new calibration: close it before calibrating.
- The calibration window stays in the foreground: move it to see the **Pan** and **Tilt** sliders.

## See also

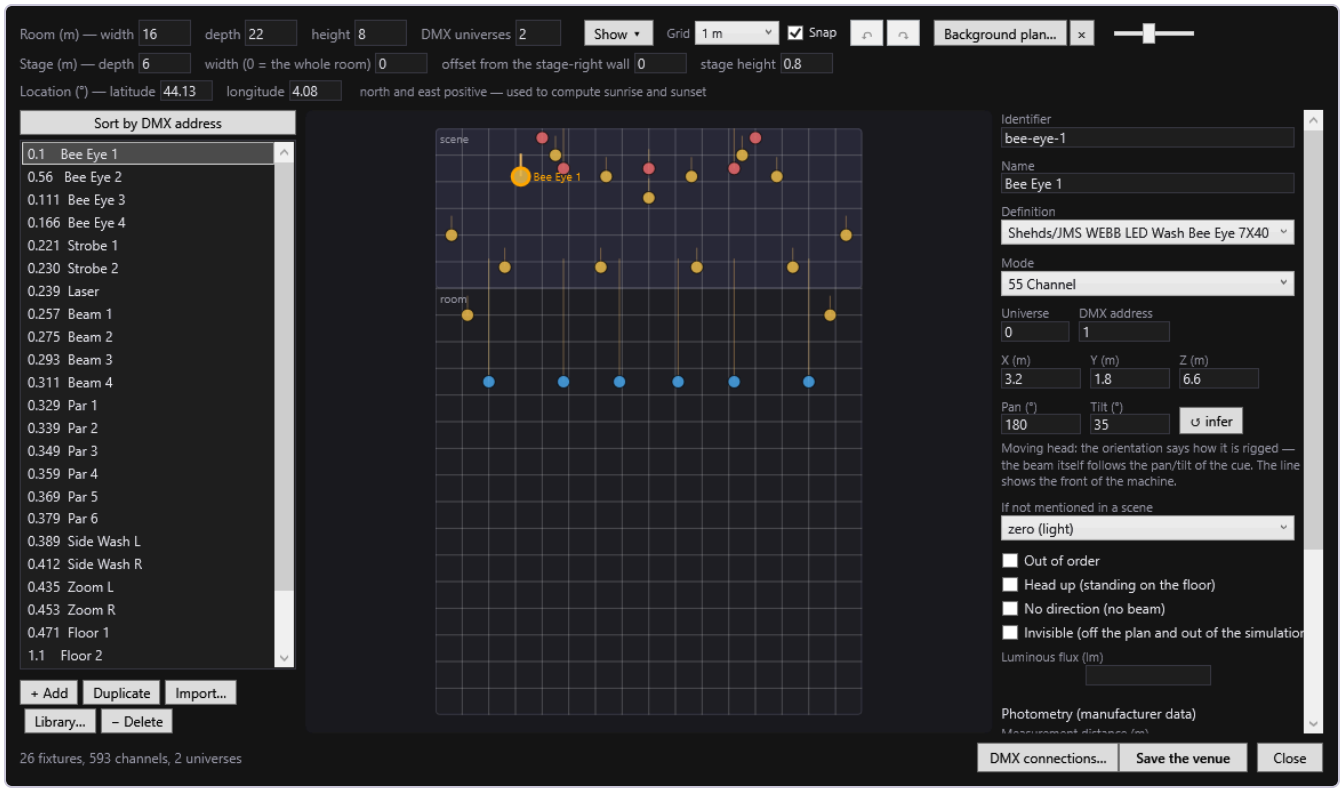
- Aiming and covering a zone with moving heads
- Fixtures, addresses, positions
- Switching, renaming, duplicating a venue
- Follow spot on a tablet
- Selecting fixtures and adjusting the light

# Fixtures, addresses, positions

This window describes the venue: its dimensions, its stage, its location on Earth and above all the list of its fixtures with their model, their DMX mode, their address and their place in the venue. This is where you "patch" the venue, that is, where you tell DreamShow which fixture responds to which DMX address, and where it is rigged. Everything else (scenes, simulation, moving-head aiming) relies on this patch.

**To open it:** menu **Venue > Fixtures, addresses, positions...**.

The window is titled "DreamShow (version) — Venue: (venue name)". It does not block the main screen: the simulation stays visible alongside while you work. If the window is already open, the menu simply brings it to the front.




The window has four areas:

- at the top, the venue bar (dimensions, stage, location, universes, display, background plan, grid, undo / redo);
- on the left, the fixture list and its buttons;
- in the centre, the venue plan seen from above;
- on the right, the sheet of the selected fixture;

- at the bottom, the status line and the **DMX connections…**, **Save the venue** and **Close** buttons.

## Important: the window works on a copy

Your changes are only applied to the venue (file, simulation, DMX output) when you click **Save the venue**. As long as you have not saved, the status line shows "— modified". If you close the window with changes pending, DreamShow asks "Venue changes not saved. Save?": **Yes** saves, **No** discards the changes, **Cancel** leaves the window open.

The copy is taken when the window opens. See "Good to know" below: whatever you change elsewhere in the venue while this window is open (a group created, a calibration, manual controls, connections set through the menu) is overwritten if you then save from this window.

## The venue bar (at the top)

### First line

- **Room (m) — width**, **depth**, **height**: the inside dimensions of the venue, in metres. The width runs from the stage-right wall (on the left as seen from the audience) to the stage-left wall; the depth runs from the upstage wall to the back wall of the house; the height is the ceiling height. Values must be greater than 0. A new venue measures 12 × 18 × 6 m. The plan is redrawn immediately. Both comma and point are accepted as the decimal separator.
- **DMX universes**: the number of DMX universes in the venue (one universe = 512 channels). This number goes up by itself if you put a fixture on a higher universe: a fixture outside the declared universes would output nothing. A new venue has 1 universe.
- **Show** ▾ : display filter by fixture category. It opens as a list of check boxes, one per category present in the venue, with two buttons, **All** (show everything) and **None** (hide everything, so you can tick back only what you want to see). The filter acts on the plan **and** on the list on the left. When categories are hidden, the button becomes "Show (N hidden) ▾ ". The selected fixture is always shown. Initially, the filter takes over the one from the main screen simulation. The categories are derived from the type of the definition: **"Lyres"** (moving heads), **"LEDs couleur"** (colour LEDs), **Dimmers**, **Scanners**, **Smoke**, **"Relais"** (relays: definitions whose model starts with "Relais") and **Others**. Each category has its own dot colour on the plan.
- **Grid**: the spacing of the grid drawn on the plan: 0.1 m, 0.25 m, 0.5 m, 1 m (default), 2 m or 5 m. The lines for each metre are bolder. This setting is not saved in the venue: it is kept as long as DreamShow stays open.
- **Snap** (ticked by default): fixtures you drag on the plan snap to the grid. When unticked, they move in free steps of 10 cm.
- ↶ (Undo, **Ctrl+Z**) and ↷ (Redo, **Ctrl+Y**): undo or redo your latest changes in this window, up to 100 steps. A burst of changes close together (a drag on the plan, typing on the keyboard) counts as a single step.

- **Background plan…**: chooses an image (plan, photo: .png, .jpg, .jpeg, .bmp, .gif) stretched over the whole venue rectangle, under the fixtures. It also appears in the main screen simulation. The image path is shown to the right of the button; if the image is in the venue's folder or one of its subfolders, the path is saved as a relative path.
- × (to the right of **Background plan…**): removes the background plan.
- Slider to the right of the path: **Background plan opacity**, from 10 % to 100 % (50 % by default).

With a background plan, the stage is no longer painted solid on the plan: it is only tinted, so the image shows through.

### Second line: the stage

- **Stage (m) — depth**: the depth of the stage, measured from the upstage wall (4.5 m for a new venue). Must be greater than 0.
- **width (0 = the whole room)**: the width of the stage. 0 = the whole width of the venue.
- **offset from the stage-right wall**: the distance between the stage-right wall and the stage-right edge of the stage (0 = stage against the stage-right wall).
- **stage height**: the height of the stage floor above the floor of the house (0.6 m for a new venue). It is used by the 3D view and by the aiming calculations: a moving head aiming at a point on the stage aims at the top of the stage floor, not at the floor of the house.

On the plan, the stage is the rectangle marked “scene”, at the top; the rest is marked “room”.

### Third line: the location

- **Location (°) — latitude** and **longitude**: the position of the venue on Earth, in decimal degrees, north and east positive (west and south negative). These two values are used only to calculate sunrise and sunset for the “sun” wait blocks in the timeline (see Waits and Architectural lighting). A new venue has the values 44.126 and 4.082.

## The fixture list (on the left)

Each line shows: status symbols, then `universe.address`, then the name. Example: `0.17 Lyre 3`.

Symbols at the start of the line:

| Symbol | Meaning |
|---|---|
| ⛔ | fixture marked **Out of order** |
| ∅ | fixture marked **Invisible** |
| ◌ | fixture marked **No direction** |

| | |
|---|---|
| ⚠ | DMX range clashing with another fixture (the line is then orange, with the tooltip “DMX range clashing with another fixture”) |

“(not placed)” is added to the name of a fixture that does not yet have a position.

Selection:

- a click selects a fixture and shows its sheet on the right;
- **Ctrl**+click and **Shift**+click select several fixtures; the sheet shows the last one clicked. A multiple selection moves as one block on the plan and is deleted in one go.

Buttons:

- **Sort by DMX address** (above the list): reorders the fixtures by universe, then by address. This order is saved with the venue.
- **+ Add**: opens the “Add a fixture — QLC+ library” picker. Type in the search field (manufacturer, model or type; several words allowed, all of them must be present); the up / down arrows move through the list without leaving the field, **Enter** or a double-click confirms. Below the list, a line recalls the type and the modes of the chosen model with their number of channels. The fixture created gets:
  - the model name;
  - an identifier taken from the model name (lower case, hyphens) followed by `-1`, `-2` …;
  - the first mode of the definition;
  - the first free address in universe 0 wide enough for all its channels;
  - a position in the centre of the venue, 3 m high.

If its category was hidden by **Show** ▾, it is shown again.

- **Duplicate**: duplicates the selected fixture **in series** (see below).
- **Import…**: imports a GDTF (.gdtf), QLC+ (.qxf) or DJworld (.fxt) fixture profile. It is converted into a QLC+ definition and joins the venue's library (see The fixture library). A message sums up the import: number of channels (including fine channels), maximum pan / tilt, modes, remarks, and the folder it was saved to. The model then appears in the **Definition** list and in the **+ Add** picker. Channels that are not recognised (macros, modes) are imported as they are, with no particular function: check them. A .ssl2 file cannot be read: use the .fxt that comes with it.
- **Library…**: opens the fixture library. When it closes, the definition lists in this window are refreshed.
- **− Delete**: removes the selected fixture (or all those in the multiple selection) from the venue, after the confirmation “Delete … from the venue?”. Deleted fixtures are also removed from the groups. (Like everything else, the deletion only becomes final when the venue is saved; ↶ undoes it.)

### Duplicating in series

The **Duplicate** button opens the small “Duplicate in series” window. It is the quick way to patch a row of identical fixtures on a bar.

- “Model: name (identifier)” recalls the fixture being copied.
- **Number of copies**: 5 by default.
- **Step X (m)**, **Step Y (m)**, **Step Z (m)**: the position offset between two copies (1 m in X by default). Positions stay inside the venue.
- **Chained DMX addresses** (ticked by default): each copy takes the address that follows the previous one (the tooltip gives the number of channels per fixture). When unticked, or if there is no more room in the universe, each copy takes the first free address found.
- **Increment the index in the name** (ticked by default): the number at the end of the name is increased (“Lyre1” gives “Lyre2”, “Lyre3”…; leading zeros are kept: “PAR 01” gives “PAR 02”); a name without a number gets an index (“Wash” gives “Wash 2”, “Wash 3”…). When unticked, each copy is called “name (copy)”.
- A preview line shows the first names and the start address.
- **Duplicate** creates the copies; **Cancel** closes without doing anything.

The copies each get a free identifier (same root as the original, followed by `-n`). They **do not take over** the calibration of the original: a moving head is calibrated from its own position.

## The venue plan (in the centre)

The plan shows the venue seen from above, with the upstage wall at the top. X runs from left (stage right) to right (stage left), Y from upstage (at the top) towards the back of the house (at the bottom).

- Each fixture is a dot in the colour of its category. A dotted dot is a fixture that has not been placed yet (it is drawn in the centre). The selection is circled in orange and its name is shown; the names of all fixtures appear when the plan is zoomed in far enough.
- A fixture that is **Out of order** or **Invisible** is drawn semi-transparent. An **Invisible** fixture does not appear on the plan at all, except when it is selected (as a “ghost”), so that you can find it.
- A dotted orange ring around a dot signals a DMX address clash.
- A tooltip on each dot gives the name, the definition, the address and any special states.

Gestures:

- **Drag a dot**: moves the fixture (X and Y). The position fields are updated. With **Snap**, the position snaps to the grid.
- **Drag a dot that is part of a multiple selection**: moves the whole group as one block, keeping the spacing; the group stops as a block against the walls.
- **Drag in an empty area**: draws a selection rectangle; all the fixtures (visible through the filter) whose dot is inside are selected.
- **Drag the ◇ handle**: points a fixture (see “Orientation” below).

### Orientation lines

- For a static fixture, a line goes from the dot to the exact spot that the simulation lights. **Solid**: the orientation has been set; **dotted**: it is "inferred": nobody has set it yet and the simulation assumes that the fixture is pointed at the stage (or, for a house COB, that it washes its wall). Inferred lines are only drawn for the selection, so as not to clutter the plan.
- For the selected fixture, a ◇ diamond ends the line. Drag it: the fixture turns towards that point (tooltip "Pull to aim: the fixture turns towards this point"). If the point is on the stage, the target is the top of the stage floor; otherwise, the floor of the house.
- For a moving head (definition with a pan), only a short stroke is drawn: it shows the front of the unit, that is, how it is rigged. The beam of a moving head follows the pan / tilt of the cue list, not how it is rigged; so there is no handle.
- Nothing is drawn for a fixture that is **No direction**, **Invisible** or **Out of order**.

## The fixture sheet (on the right)

The column scrolls if it does not fit vertically. All fields apply to the selected fixture.

- **Identifier**: the unique technical name of the fixture in the venue (for example `lyre-3`). It is the lasting key: scenes, groups and calibrations refer to it. An identifier that is already in use is refused ("⚠ identifier "…" already in use — not applied"); the fixture keeps the old one until the text is unique again. When you change an identifier, the venue's groups follow, **but not the shows**: scene settings that targeted the old identifier no longer find the fixture.
- **Name**: the name shown everywhere (list, simulation, selection). Unlike the identifier, it can be repeated.
- **Definition**: the fixture model, in the form "Manufacturer/Model", chosen from the library. Changing the definition resets the mode to the first mode of the new model.
- **Mode**: the DMX mode (number and order of channels) **set on the fixture itself**. It must be the same as on the unit, otherwise the channels will be shifted.
- **Universe**: the fixture's DMX universe, **counted from 0**: 0 is the first universe, which is called **U1** in the DMX connections window; 1 corresponds to U2, and so on. If you enter a universe beyond the declared number, the **DMX universes** field at the top goes up by itself.
- **DMX address**: the first channel of the fixture in its universe, from 1 to 512, as on the fixture's display. A value outside 1–512 is ignored.
- **X (m)**, **Y (m)**, **Z (m)**: the position of the fixture in metres. X = across from the stage-right wall, Y = depth from the upstage wall, Z = height from the floor of the house. Dragging on the plan fills in X and Y; Z is typed in (3 m for a fixture that has just been added).

### Orientation

- **Pan (°)**: rotation around the vertical axis, in degrees. 0 = the front of the fixture facing the audience, 180 = facing upstage. The line on the plan shows the result.
- **Tilt (°)**: tilt of the beam. 0 = horizontal, 90 = straight down, negative = upwards.

- ↺ **infer**: clears the set orientation. The simulation goes back to its assumption (fixture pointed at the stage, house COB washing its wall). The button is only active if an orientation has been set.

When the orientation is not set, both fields show the angles of the assumption: the first change therefore starts from what is drawn, without a jump. Below the fields, a help line explains the situation:

- "Orientation set: the simulation lights the point at the end of the line."
- "Orientation inferred (dotted line): … Pull the ◇ handle to fix it."
- for a moving head: the orientation says how it is rigged; the beam follows the pan / tilt of the cue list;
- for a fixture that is **No direction** or **Invisible**, the orientation fields are disabled and the help line says so.

For a moving head, the orientation also matters for aiming: the **Aim** and **Zone** tools take it into account (see Aiming and covering a zone with moving heads).

### Behaviour and attributes

- **If not mentioned in a scene**: what the fixture does when the final state of a scene does not set it.
  - **zero (light)** (default): it goes to zero, like any fixture.
  - **hold**: it keeps the values from the previous scene. In addition, the general blackout does not reset its channels to zero. This is the setting for devices that move physically or that must not switch off: hoist, power supplies.
- **Out of order**: the fixture is out of service. It disappears from the simulation (2D and 3D) and is drawn semi-transparent on this plan, but its patch, its groups and its DMX output do not change: unticking the box after repair is enough. The **Zone** tool ignores moving heads that are out of order.
- **Head up (standing on the floor)**: for a moving head standing base down, head up (on the floor, on a riser), instead of being hung head down. Aiming and the simulation take the flip into account; the front face stays turned towards the audience.
- **No direction (no beam)**: smoke or bubble machine, LED strip, house lights, power supply… No orientation on the plan, no beam in the simulation. The fixture stays visible and lights up when it emits.
- **Invisible (off the plan and out of the simulation)**: relay, power supply, equipment whose DMX matters but not its place on the stage. Nothing is drawn, neither on the plan nor in the simulation. Patch, groups and DMX output unchanged. The selected fixture is still shown as a ghost so it can be found.

Fixtures that are **No direction**, **Invisible** or **manual** are never switched on by automatically written moods (first show wizard).

The **manual** attribute is not set here: a fixture becomes manual in **Venue > Manual controls…** (see Manual controls).

### Photometry

These fields make the simulation more realistic (effect of the zoom, the distance, the angle of incidence). They change nothing in the DMX output.

- **Luminous flux (lm)**: the flux actually emitted by the fixture, in lumens. Empty = the value from the library, or failing that an estimate based on the power.
- **Photometry (manufacturer data)**: readings from the data sheet, at both ends of the zoom:
  - **Measurement distance (m)** (5 m if empty);
  - **Wide beam (°)** and **lux at this distance**;
  - **Narrow beam (°)** and **lux at this distance**.

Example given in the tooltip: 545 lux at 5 m at 60°, 14335 lux at 5 m at 5°. The readings are only used if they are complete and consistent (both lux values and the narrow angle greater than 0, the wide angle larger than the narrow angle). They then replace the estimate based on the flux **and** the angles from the library.

Below these fields, a line shows what the simulation is really using: the manufacturer photometry, the flux entered, the flux from the library, an estimate based on the power (with an invitation to enter the real value), or “No photometric data” (the simulation then keeps neutral lighting for this fixture).

At the bottom of the column, a reminder: drag a fixture on the plan to place it, and the meaning of **hold**.

## The status line and saving (at the bottom)

The status line sums up the venue: “N fixtures, N channels, N universes”, followed by “— modified” if there are unsaved changes. If there are problems, it turns orange and lists the first three after ⚠ (the number of further problems is given in brackets).

The checks cover:

- an empty or duplicate identifier;
- an unknown definition (missing from the library) or an unknown mode;
- an address that pushes the fixture outside the 1–512 range;
- two fixtures whose channels overlap in the same universe (“channel N (universe U): A and B overlap”).

Buttons:

- **DMX connections…**: sets up the universes and the output interfaces of this venue (see DMX connections). Here, the connections are saved **with the venue**, when you click **Save the venue**.
- **Save the venue**: applies and saves. An empty or duplicate identifier, or an unknown definition or mode, **prevents** saving (message “Cannot save:” with the list). Overlapping addresses ask for confirmation (“DMX ranges overlap: … Save anyway?”). After saving, the main screen reloads the patch, the selection tree and the simulation, and restarts the DMX output if the number of universes or the connections have changed. “Venue saved” appears at the top of the main screen.
- **Close**: closes the window (with the save question if there are changes left).

## Groups

Groups (for example “Moving heads”, “Front”, “Back”) let you set several fixtures at once in scenes. They belong to the venue, but they are not created in this window: they are created in the **Libraries** panel of the main screen.

1. Select the fixtures for the future group (the “Selection — groups and fixtures” tree, clicking dots in the **Venue** panel, or a selection rectangle).
2. In the **Libraries** panel, **Groups** section, click **+ group (selection)**.
3. Type the group name. A name already used by a group or by a fixture identifier is refused (“That name already exists.”).

The group is saved in the venue straight away. Groups appear in the order in which they were created; there is no command to reorder them. To delete a group: right-click its name in the **Groups** list, **Delete the group**; the fixtures remain, only the show entries that targeted that group no longer target anything. See also Libraries.

When you delete a fixture from the venue, it is removed from all groups. When you change its identifier, the groups follow.

## Step by step

### Patching a row of 6 identical LED PARs on a front-of-house bar

1. Open **Venue > Fixtures, addresses, positions…**.
2. Click **+ Add**, type the manufacturer or the model, choose it, **Add**.
3. In the sheet, choose the **Mode** set on your PARs, check **Universe** (0 = U1) and **DMX address** (for example 1).
4. Place the first PAR: drag its dot on the plan to the stage-right end of the bar, then type its height in **Z (m)** (for example 6).
5. Click **Duplicate**. Set **Number of copies** to 5, **Step X (m)** to the gap between two PARs (for example 1.5), leave **Chained DMX addresses** and **Increment the index in the name** ticked,

then **Duplicate**.
6. Check the status line: no ⚠.
7. If necessary, point each PAR: select it and drag its ◇ handle onto the area it must light.
8. Click **Save the venue**.
9. Physically set each PAR to the address shown in the list.
10. Create a “Front” group: select the 6 PARs in the main screen, then **+ group (selection)**.

### Matching the venue to an architect's plan

1. Measure (or read off the plan) the inside width and depth, and type them into **Room (m) — width** and **depth**.
2. Click **Background plan…** and choose an image of the plan, cropped exactly to the inside walls (the image is stretched over the whole venue rectangle).
3. Adjust the opacity with the slider so the dots are clearly visible.
4. Adjust **Stage (m) — depth**, **width**, **offset from the stage-right wall** and **stage height** until the “scene” rectangle falls on the drawn stage.
5. Place the fixtures on the plan, then **Save the venue**.

### Resolving an address clash

1. Find the orange lines marked ⚠ in the list (and the orange rings on the plan). The status line gives the channel and the two fixtures involved.
2. Select one of the two and correct its **DMX address** (or its **Mode**, if it is longer than expected).
3. **Sort by DMX address** helps you see how the addresses follow on.
4. When no ⚠ remains, **Save the venue**.

## Good to know

- **Universes counted from 0**: the **Universe** field and the list (`0.17`) start at 0, whereas the DMX connections window and the manual controls show U1, U2… for the same universes. Universe 0 = U1.
- **The mode must be the one on the unit.** If the fixture is set to 8 channels and the venue to 12, everything is shifted: wrong colours, moving heads flying off in all directions.
- **Changing an identifier** breaks the link with the scenes already written for that fixture (the groups, however, follow). Change the **Name** instead.
- **Copy taken on opening**: the window works on a copy of the venue taken when it opens, and **Save the venue** rewrites the whole venue with that copy. If, while it is open, you create a group, calibrate a moving head, set manual controls or change the connections through the menu, saving from this window afterwards undoes those changes. Close the window before these operations, or reopen it afterwards.
- If the **Manual controls** window is open when you save the venue here, it updates itself (addresses, list of manual fixtures, values).

- A fixture that is **Out of order** still outputs DMX: only the display changes.
- **Duplicate** does not copy moving-head calibration.
- The grid and snapping are working settings, not venue data.
- The simulation needs positions: a “(not placed)” fixture is drawn in the centre.

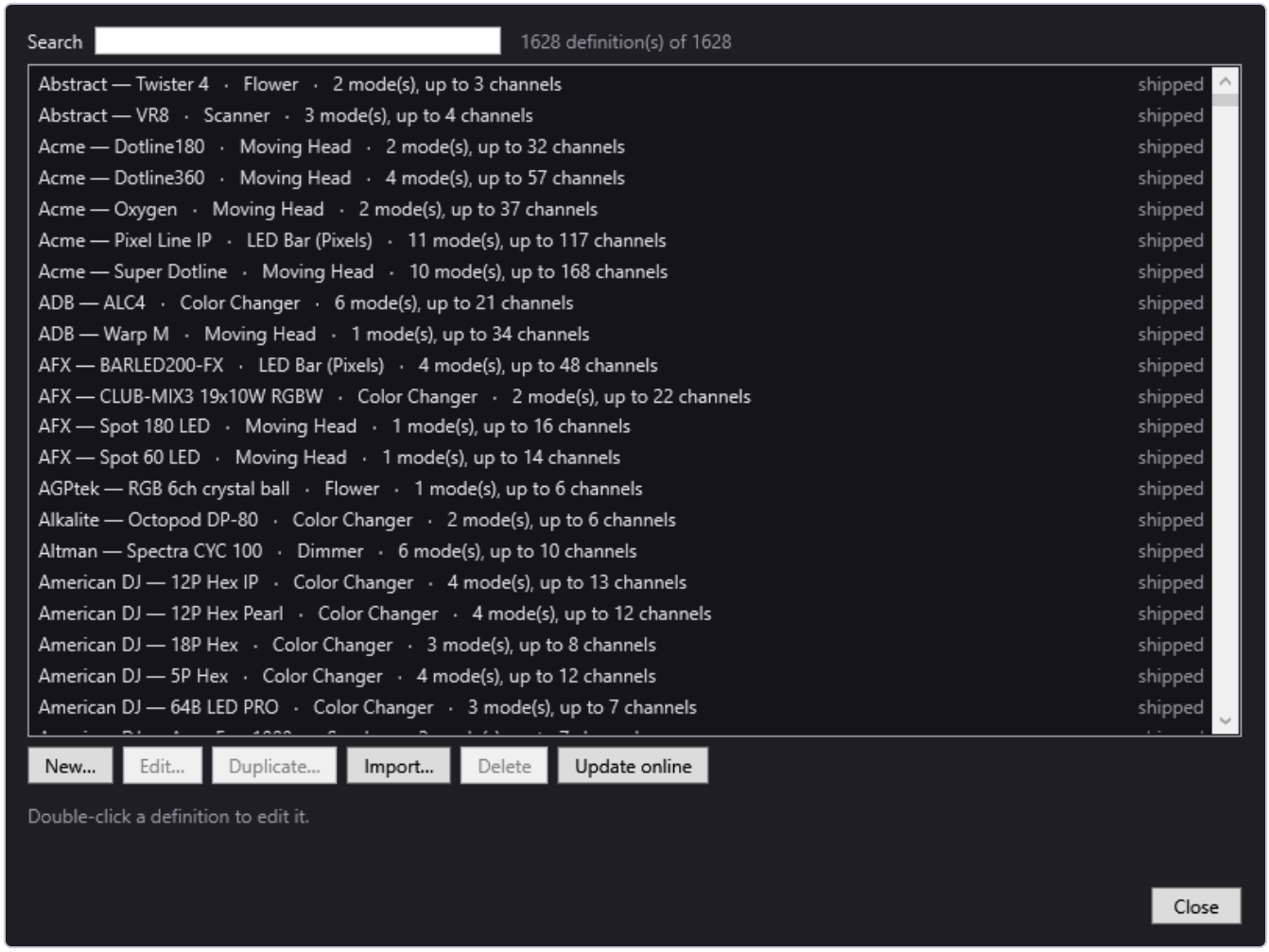
## See also

- The fixture library
- DMX connections
- Manual controls
- Aiming and covering a zone with moving heads
- Moving-head calibration
- Declaring your fixtures (venue wizard)
- Switching, renaming, duplicating a venue
- 2D and 3D simulation
- Libraries

# The fixture library

The library contains the fixture **definitions**: for each model, its DMX channels, their functions, its modes, its cells and some physical data. When you add a fixture to the venue, you choose its definition here. DreamShow comes with more than 1,600 definitions in QLC+ format; you can import, create, edit and update them online.

**To open it:** menu **Venue > Fixture library...**, or the **Library...** button in the **Venue > Fixtures, addresses, positions...** window.




## Where the definitions come from

The library is a folder (by default the `fixtures` folder next to the venue files), shared by all the venues that use it. Each definition has an origin, shown on the right of its line:

| Origin shown | Where it comes from |
|---|---|

| | |
|---|---|
| **shipped** | installed with DreamShow |
| **online** | downloaded with **Update online** (“online” folder) |
| **imported** | imported from a GDTF, QLC+ or DJworld file (“imported” folder) |
| **created / edited** | created or edited in the DreamShow editor (“custom” folder) |

When two definitions have the same manufacturer and the same model, the most “personal” one wins, in this order: created / edited, then imported, then online, then shipped. The definition that is overridden is not erased: it is only hidden, and it comes back if you delete the one that replaces it.

A **shipped** or **online** definition is never modified: its edited version is saved separately, in the “custom” folder, and replaces it.

## The Fixture library window

- **Search**: filters the list. Type one or more words; a definition is kept if each word appears in its manufacturer / model or in its type (for example “Moving Head”). On the right, the counter “N definition(s) of M”. The list shows at most 2,000 lines: narrow down the search if necessary.
- **The list**: one line per definition: “Manufacturer — Model · Type · N mode(s), up to M channels”, with the origin on the right (in colour if it is not “shipped”). A double-click opens the definition in the editor.
- **The information line** (below the buttons): for the selected definition, its modes with their number of channels (and cells), the file, and “Replaces another definition with the same name (deleting this one brings it back).” where applicable. With nothing selected: “Double-click a definition to edit it.”

Buttons:

- **New…**: creates a definition from scratch in the editor.
- **Edit…**: opens the selected definition in the editor. For a shipped or online definition, the edited version is saved separately and replaces it.
- **Duplicate…**: opens a copy of the selected definition (model followed by “(copy)”), to make another one from it: another model, another mode.
- **Import…**: imports one or more profiles (multiple selection possible): GDTF (.gdtf), QLC+ (.qxf) or DJworld (.fxt). A report lists each file: ✓ with the number of modes and the remarks, or ✗ with the reason for the failure.
- **Delete**: only active for an **imported** or **created / edited** definition. After confirmation, the file is erased. If it replaced a definition with the same name, that one takes its place again; otherwise, the fixtures that used it have no definition until another one is chosen.
- **Update online**: downloads the new or corrected definitions published for DreamShow (Internet connection required). Progress is shown in the information line, then a summary: number of definitions published, new, updated, already up to date, withdrawn, and any

problems. Each downloaded file is checked (fingerprint and reading) before anything is replaced. A definition withdrawn from the published list is deleted from the “online” folder only; your imported or created definitions are never touched.
- **Close**.

Any change (creation, edit, import, deletion, update) is taken into account immediately: the venue, the simulation and the show are recalculated with the new definitions.

## Importing

- **GDTF (.gdtf)**: the manufacturers' format (manufacturers' websites, GDTF Share). Each DMX mode is converted; 16- or 24-bit channels become a main channel followed by “fine” channels; wheel colours are taken over; the shutter channel is named “Shutter” with its “open” range. Only the first DMX break is imported: anything left out is reported in the remarks.
- **QLC+ (.qxf)**: the file is checked, then copied into the “imported” folder.
- **DJworld (.fxt)**: converted into a QLC+ definition. The .ssl2 file that sometimes comes with these profiles cannot be read: use the .fxt.

Imports are stored in the library's “imported” folder, by manufacturer. Then check the definition in the editor: channels that are not recognised (macros, modes) are imported as they are, with no particular function.

## The definition editor

It opens with **New…**, **Edit…**, **Duplicate…** or a double-click. Its title is “New fixture definition”, “Definition “model”” or “Copy of “model””.

A new definition starts from an example: manufacturer “My brand”, model “New fixture”, type “Color Changer”, one mode “4 channels” (Dimmer, Red, Green, Blue).

### The header

- **Manufacturer** and **Model**: required. Together, they form the name of the definition (“Manufacturer/Model”) that venues use. Do not change this name lightly: the fixtures in your venues refer to it.
- **Type**: editable list: Moving Head, Color Changer, LED Bar (Pixels), LED Bar (Beams), Dimmer, Strobe, Scanner, Smoke, Hazer, Effect, Laser, Flower, Other. Empty = Other. The type puts the fixture into a simulation category (Moving Head = **“Lyres”** (moving heads), Color Changer = **“LEDs couleur”** (colour LEDs), Dimmer = **Dimmers**, Scanner = **Scanners**, Smoke = **Smoke**, the rest = **Others**).
- **Max pan (°)** and **Max tilt (°)**: the total travel of pan and tilt, in degrees (for example 540 and 270), taken from the fixture's manual. A **Max pan** greater than 0 makes the definition a

moving head for DreamShow: this is what makes it usable by **Aim**, **Zone**, trajectories and calibration. Wrong values throw off all the aiming.
- **Beam (°) from … to …**: the beam angles at both ends of the zoom. With a zoom channel, they let the **Zone** tool (and **zoom on aim**) set the beam width; they are also used by the simulation.
- **Power (W)** and **Luminous flux (lm)**: data from the data sheet, used by the simulation to estimate the illuminance. The real flux of a particular fixture can also be entered in the venue.

### Modes

On the left, **DMX modes** lists the modes with their number of channels. Click a mode to show its channels.

- **Mode name**: renames the selected mode. This is the name you will choose in the venue's **Mode** field; give it a meaningful name (for example “14 channels”). Two modes cannot have the same name.
- **+ Mode**: adds an empty mode (“Mode N”).
- **Duplicate**: copies the selected mode (name followed by “(2)”), handy for an extended mode close to another one.
- **Delete**: deletes the selected mode (the last mode cannot be deleted).

### The channels of the mode

On the right, a table with one line per channel, in DMX order (the **#** column gives the channel number within the mode, starting from 1).

- **Channel name**: the name of the channel. It matters: DreamShow recognises certain names (see “Naming rules”). Two channels in the same mode cannot have the same name.
- **Function**: what the channel does. This is what tells DreamShow how to drive it. Possible choices:

| Function | Use |
|---|---|
| Master dimmer | intensity of the whole fixture |
| Dimmer (cell) | intensity of one cell |
| Red, Green, Blue, White, Amber, UV, Lime, Cyan, Magenta, Yellow | colour components |
| Pan, Tilt | moving-head movements |
| Pan/tilt speed | motor speed |
| Shutter / strobe | shutter, strobe |
| Colour wheel | colour wheel |

| | |
|---|---|
| Colour macros | preprogrammed colours |
| CTO correction | colour temperature |
| Gobo, Gobo rotation | gobos |
| Prism | prism |
| Zoom, Focus, Iris, Frost | beam |
| Effect / other | everything else |
| Maintenance (reset, lamp) | maintenance commands |
| Fine byte of the previous channel | precision byte (16-bit) of the channel just above |
| Reserved | unused channel |

An imported definition may contain a QLC function that the list does not know: it is shown as “QLC preset “…”” and kept as it is. Changing the function also changes the channel name as long as you have not typed one yourself. Choosing **Shutter / strobe** always names the channel “Shutter”.

- **Cell**: the number of the cell (pixel, lens) the channel belongs to; empty = the whole fixture. Channels with the same number form a cell. If only one number is used, the fixture is treated as having no cells.
- **Ranges (N)…**: opens the value ranges of the channel (see below). Inactive for a fine byte.
- ↑ / ↓: moves the channel up or down one step in DMX order.
- ✕: deletes the channel.

Below the table:

- **+ Channel**: adds an “Effect” channel at the end of the mode.
- **+ Cells…**: adds a series of cells in one go (“Add cells” window):
  - **Number of cells** (8 by default);
  - **Composition**: RGB, RGBW, RGBA, RGBWA, RGBWAUV, W or RGB + Dimmer;
  - **16-bit colours (fine byte after each colour)**.

The channels are created and numbered after the existing cells: “Red 1”, “Green 1”, “Blue 1”, “Red 2”…, each in its own cell.

For a fine byte, the name is set automatically (“previous channel name” followed by “fine”) and it takes the cell of the previous channel. A fine byte must directly follow its channel.

### Value ranges

The “Value ranges — channel name” window describes what each slice of values (0 to 255) of the channel does.

- **From** and **To**: the limits of the range (0 to 255).
- **Label**: description (empty = the channel name).
- **Type**:
  - **Normal**;
  - **Shutter open**: the range DreamShow sends when light must come out;
  - **Shutter closed**;
  - **Strobe**;
  - **Colour**: for a colour wheel or a macro; the **Colour** box becomes active.
- **Colour**: the colour of the range in hexadecimal (`#FF8800`; the `#` is added if you forget it), with a preview. It is used by the simulation.
- ✕: deletes the range.
- **+ Range**: adds a range that starts just after the last one.
- **OK** confirms, **Cancel** discards. Ranges are sorted by start value.

Values that are not covered are read as “no function”.

### Saving

At the bottom of the editor, a line recalls where the definition will be saved. **Save** checks the definition and then writes it; **Cancel** closes without writing anything.

Saving is refused, with the list of problems, if:

- the manufacturer or the model is empty;
- a mode has no name, two modes have the same name, or a mode has no channel;
- two channels in a mode have the same name (the message suggests numbering them: “Red 1”, “Red 2”);
- a channel name ends in “fine” without being a fine byte;
- a fine byte does not follow its channel;
- a range starts after its end;
- a mode has a **Shutter / strobe** channel but no channel named “Shutter”.

The definition is saved in the library's “custom” folder (`custom\Fabricant\Modèle.qxf`). It appears in all the venues that use this library. If you rename a definition that you created, the old file is deleted to avoid a duplicate.

## Naming rules

DreamShow recognises certain channels by their **name**; follow these rules so that a fixture lights up and takes its colour correctly:

- **Shutter**: the shutter channel must be called exactly “Shutter”. Otherwise DreamShow does not know how to open it and the fixture may stay dark. For a fixture without a “Shutter” channel,

DreamShow tries to open a channel whose name starts with “Strobe”; declare its **Shutter open** range so that it finds the right value.

- **Cells**: the colour channels of each cell are numbered: “Red 1”, “Green 1”, “Blue 1”, “White 1”, then “Red 2”… (“Red1” and “Red 1 Dimmer” are also recognised). This is what lets you set cells one by one in the scene and animate them in trajectories. **+ Cells…** applies this rule by itself.
- **Fine bytes**: their name ends in “fine”, and that ending is reserved for them.
- For a hoist driven by the manual controls, the “Position” and “Speed” channels (and “Position fine”) make the hoist controls appear.

## Step by step

### Creating the definition of a fixture missing from the library

1. Open the fixture's manual at the page with its DMX table.
2. **Venue > Fixture library…**, then **New…**.
3. Fill in **Manufacturer**, **Model** and **Type**. For a moving head, fill in **Max pan (°)** and **Max tilt (°)**; if it has a zoom, **Beam (°) from … to …**.
4. Rename the “4 channels” mode after the mode set on the unit (for example “9 channels”).
5. For each channel in the table, in order: **+ Channel** if needed, choose the **Function**, correct the **Channel name**. For a 16-bit channel, add a channel with the function **Fine byte of the previous channel** just after it.
6. For the shutter: function **Shutter / strobe** (the name becomes “Shutter”), then **Ranges…**: declare the **Shutter open** range (for example from 0 to 7 if the table says so), the **Strobe** ranges, and so on.
7. For a colour wheel: **Ranges…**, one range of type **Colour** per colour, with its hexadecimal colour.
8. For another mode of the unit: **Duplicate** the mode, rename it, add or remove channels.
9. **Save**. The model is now offered by **+ Add** in the venue.

### Describing an LED bar with 8 RGBW cells

1. **New…**, manufacturer, model, type “LED Bar (Pixels)”.
2. In the mode, keep a **Master dimmer** channel if needed, and delete the example Red / Green / Blue channels (✕).
3. **+ Cells…**: **Number of cells** 8, **Composition** RGBW, OK.
4. Check that the channels are called “Red 1” … “White 8” with cell numbers 1 to 8, then **Save**.

### Correcting a shipped definition

1. Search for it, select it, **Edit…**.
2. The bottom line indicates that your changes will be saved separately: the original stays intact.

3. Make the correction (for example, name the shutter channel “Shutter”), **Save**. The definition's line now shows “created / edited”.
4. To go back to the original: select it, **Delete**.

### Importing a manufacturer's GDTF profile

1. Download the fixture's .gdtf file.
2. **Import…**, choose the file.
3. Read the report (modes, remarks about what was not imported).
4. Open the imported definition with **Edit…** to check the channel functions and the shutter.

## Good to know

- The library is common to all the venues that share it: an edited definition changes all those venues.
- A definition deleted while fixtures use it makes those fixtures “unknown definition” in the venue window (saving the venue is then blocked until you choose another definition).
- **Update online** never replaces your imported or created definitions: they keep priority.
- Changing the manufacturer or model of a definition changes its name: fixtures that used the old name no longer find it.
- The editor cannot undo changes one at a time: if you make a mistake, **Cancel** closes without saving anything.

## See also

- Fixtures, addresses, positions
- Declaring your fixtures (venue wizard)
- Selecting fixtures and adjusting the light
- Aiming and covering a zone with moving heads
- Manual controls
- Glossary

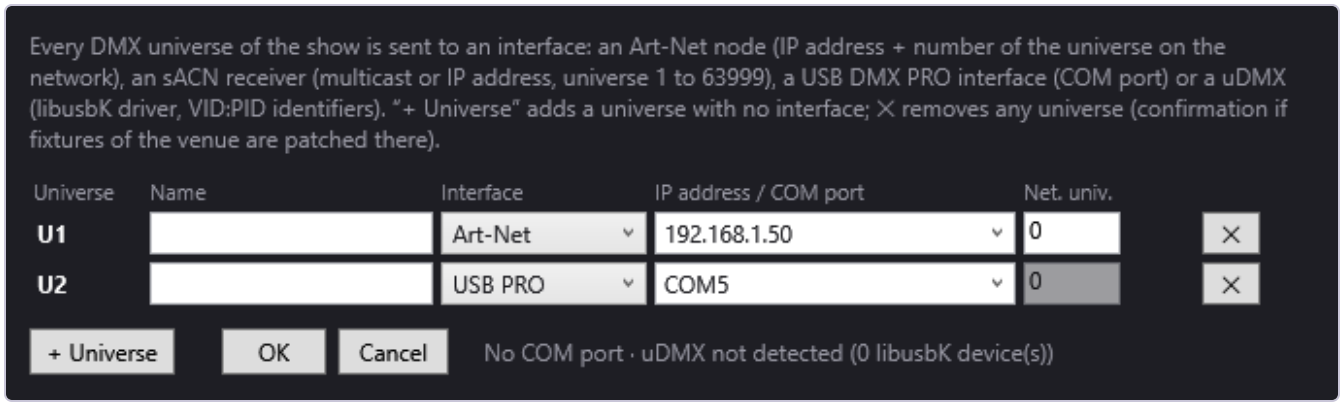
# DMX connections

The DMX connections window tells DreamShow **where the venue's DMX universes go out**: an Art-Net node on the network, an sACN receiver, a USB DMX PRO interface or a uDMX interface. Connections belong to the venue: they are saved in its file, and switching venues also switches the outputs.

**To open it:**

- menu **Venue > DMX connections...**;
- the **Connections...** button at the top right of the main screen;
- the **DMX connections...** button at the bottom of the **Venue > Fixtures, addresses, positions...** window.

The window is titled “DMX connections — (venue name)”.




## Basics

- A DMX **universe** carries 512 channels. A small installation fits in one universe; beyond that, you need several.
- Each universe of the venue is sent to **one** interface. Several universes can share the same Art-Net node or the same sACN receiver (each with its own network universe number). A USB interface has only one cable: one universe per USB interface.
- Universes are numbered **U1, U2, U3...** in this window. Caution: in the **Fixtures, addresses, positions...** window, a fixture's **Universe** field starts at 0. A fixture's universe 0 = **U1** here, universe 1 = **U2**, and so on.

## The list of universes

One line per universe, with the columns:

- **Universe**: U1, U2… The tooltip says “Universe used by the venue patch” or “Extra universe”.
- **Name**: a free name to help you find your way (for example “Front”, “Ceiling”, “Stage-left node”).
- **Interface**:
  - **Art-Net**: an Art-Net node on the network (UDP protocol, port 6454);
  - **sACN (E1.31)**: an sACN receiver, by multicast or at an IP address (port 5568);
  - **USB PRO**: a USB interface of the Enttec DMX USB PRO type, on a COM port;
  - **uDMX**: a uDMX interface (libusbK driver);
  - **None**: the universe exists in the venue but goes out nowhere (handy for patching before you have the equipment).
- **IP address / COM port**: editable list (you can type a value or pick one of those offered):
  - in Art-Net: the node's IP address. Offered values are 127.0.0.1 (the PC itself, for other software on the same computer), 2.255.255.255, 10.255.255.255 and 255.255.255.255 (broadcast addresses to a whole network);
  - in sACN: `multicast` (sending to the universe's multicast group, the usual case) or the IP address of a receiver; 127.0.0.1 is offered;
  - in USB PRO: the interface's COM port; the COM ports present are offered;
  - in uDMX: the USB identifiers in the form VID:PID; the standard value `16C0:05DC` is offered;
  - with **None**: the field is disabled.
- **Net. univ.**: the universe number on the network, to be matched with the setting of the node or receiver. In Art-Net: 0 to 32767. In sACN: 1 to 63999. Disabled for USB PRO, uDMX and None.
- ✕: removes the universe. If fixtures of the venue are patched on it, a confirmation reminds you: “Fixtures of the venue are patched on U… Remove that universe anyway? They will no longer go out anywhere.”

When you change the interface, DreamShow suggests a suitable address: it keeps an IP address already entered (Art-Net, sACN), otherwise it puts 127.0.0.1 (Art-Net) or `multicast` (sACN); it suggests the first COM port present (or COM5) for USB PRO, and `16C0:05DC` for uDMX. In sACN, if the network universe number is not valid, it suggests the universe number (U3 → 3).

## The buttons and the information line

- **+ Universe**: adds a universe at the end, with the interface **None**. Then choose its interface if needed.
- **OK**: checks, saves and applies.
- **Cancel**: closes without changing anything.
- To the right of the buttons: the COM ports present (“Ports COM : COM3, COM5” (COM ports) or “No COM port”), then the uDMX status: “uDMX detected”, “uDMX not detected (N libusbK device(s))” or “libusbK.dll missing (uDMX driver not installed)”. This line is read when the window opens: connect the interface before opening it.

When you click **OK**, DreamShow refuses (and lets you correct):

- an invalid IP address in Art-Net or sACN ("U…: invalid IP address");
- a network universe out of range ("invalid network universe number (0 to 32767)" in Art-Net, "invalid sACN universe (1 to 63999)" in sACN);
- a USB interface without a COM port ("give the COM port of the USB interface").

If several universes are sent to the same USB interface, DreamShow warns that they will write in turn on the same cable and asks whether to continue.

## What happens after OK

- From the **Venue > DMX connections…** menu or the **Connections…** button: the connections are saved straight away in the venue file, and the DMX output restarts on the new interfaces. The Log records "Connections saved in …".
- From the **Fixtures, addresses, positions…** window: the connections are part of the venue changes; they are saved and applied when you click **Save the venue**.

If an interface does not open (COM port missing, driver missing, impossible address…), a "DMX connections — interfaces that failed" window lists the universes concerned with the reason. The other universes go out normally: the show carries on with the ones that work.

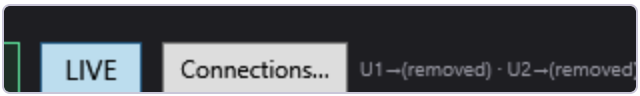
A venue whose connections have never been set uses the settings saved on this PC by older versions; failing that, each patched universe goes out in Art-Net to 127.0.0.1, with network universe 0 for U1, 1 for U2, and so on. Open the window and click **OK** to fix the connections in the venue.

## Output status in the main screen

At the top right of the main screen:

- **LIVE** (on by default): when on, the selected scene is sent live to the interfaces (Art-Net, USB, uDMX). When off, no frame goes out any more: only the simulation continues. Handy for programming in peace without the stage moving. This button has no effect in show mode (output is always active during the performance). When you switch **LIVE** back on, DreamShow jumps to the selected scene. The Log records "LIVE off: no frame is sent to the interfaces any more" or "LIVE: frames sent to the interfaces again". During Visualiser mode, DreamShow stays silent, whatever the state of **LIVE**.
- **Connections…**: opens this window.
- **The grey text to the right of Connections…** sums up the outputs in service, universe by universe, for example `U1→Art-Net 192.168.1.50/0 · U2→USB COM5`. The tooltip repeats this text with any errors. Possible signs:
  - `U2→—`: universe without an interface (**None**);
  - `U1→ÉCHEC` (shown as `U1→FAILED` in English): the interface could not be opened;

- `U3→(retiré)` (`U3→(removed)`): universe used by fixtures but removed from the connections;
- “no DMX output”.




## Step by step

### Output to an Art-Net node

1. Put the PC and the node on the same network (for example PC at 192.168.1.10, node at 192.168.1.50, mask 255.255.255.0).
2. Note the universe number set in the node (often 0 for its first port).
3. **Venue > DMX connections...**. On the **U1** line: **Interface** = **Art-Net**, **IP address / COM port** = 192.168.1.50, **Net. univ.** = the number you noted.
4. **OK**. Check the summary at the top right: `U1→Art-Net 192.168.1.50/0`.
5. If Windows asks you to allow DreamShow on the network, accept.

### Output to a USB DMX PRO interface

1. Connect the interface **before** starting DreamShow, and check that its driver (FTDI for Enttec-type interfaces) is installed.
2. **Venue > DMX connections...**: the bottom line must list its COM port.
3. On the **U1** line: **Interface** = **USB PRO**, **IP address / COM port** = the interface's COM port.
4. **OK**.

### Adding a second universe

1. In **Fixtures, addresses, positions...**, put the fixtures concerned on **Universe** 1 (which will be **U2**).
2. **Venue > DMX connections...**: if U2 does not appear, **+ Universe**.
3. Choose its interface (for example Art-Net, same node as U1 with **Net. univ.** 1, or the node's second port).
4. **OK**.

## Troubleshooting: nothing comes out

1. **LIVE** must be on (outside show mode).
2. **What DreamShow is sending**: open **DMX > DMX values...**. If the values move, DreamShow is sending; the problem lies between the PC and the fixtures. See DMX values and DMX faders.

3. **The output summary**: a `ÉCHEC`, a `—` or a `(retiré)` on the universe concerned explains the silence. Read the tooltip and the Log.
4. **In Art-Net**: same network for the PC and the node, correct network universe number, Windows firewall allowing DreamShow.
5. **In sACN**: correct universe number (1 to 63999); if the network does not let multicast through, enter the receiver's IP address.
6. **In USB**: interface connected before starting, driver installed, correct COM port, only one universe per interface.
7. **In uDMX**: the bottom line must say “uDMX detected”; otherwise, install the libusbK driver.
8. **The right universe**: remember that a fixture's universe 0 goes out on **U1**.
9. **On the fixture side**: DMX address and mode (number of channels) identical to those in the venue, DMX cable in good condition, terminator at the end of the line.
10. **To test a single fixture**: **DMX > DMX faders…** forces channels by hand.

## Good to know

- Connections follow the venue: when you switch venues, DreamShow restarts on the outputs of the new venue.
- A universe can exist without an interface (**None**): useful for preparing a patch before you have the equipment.
- It is possible to remove a universe on which fixtures are patched: they no longer go out anywhere, and the summary shows this with `(retiré)`.
- The **Fixtures, addresses, positions…** window works on a copy of the venue: if you change the connections through the menu while it is open, then save the venue from it, it is that window's own connections that are saved.

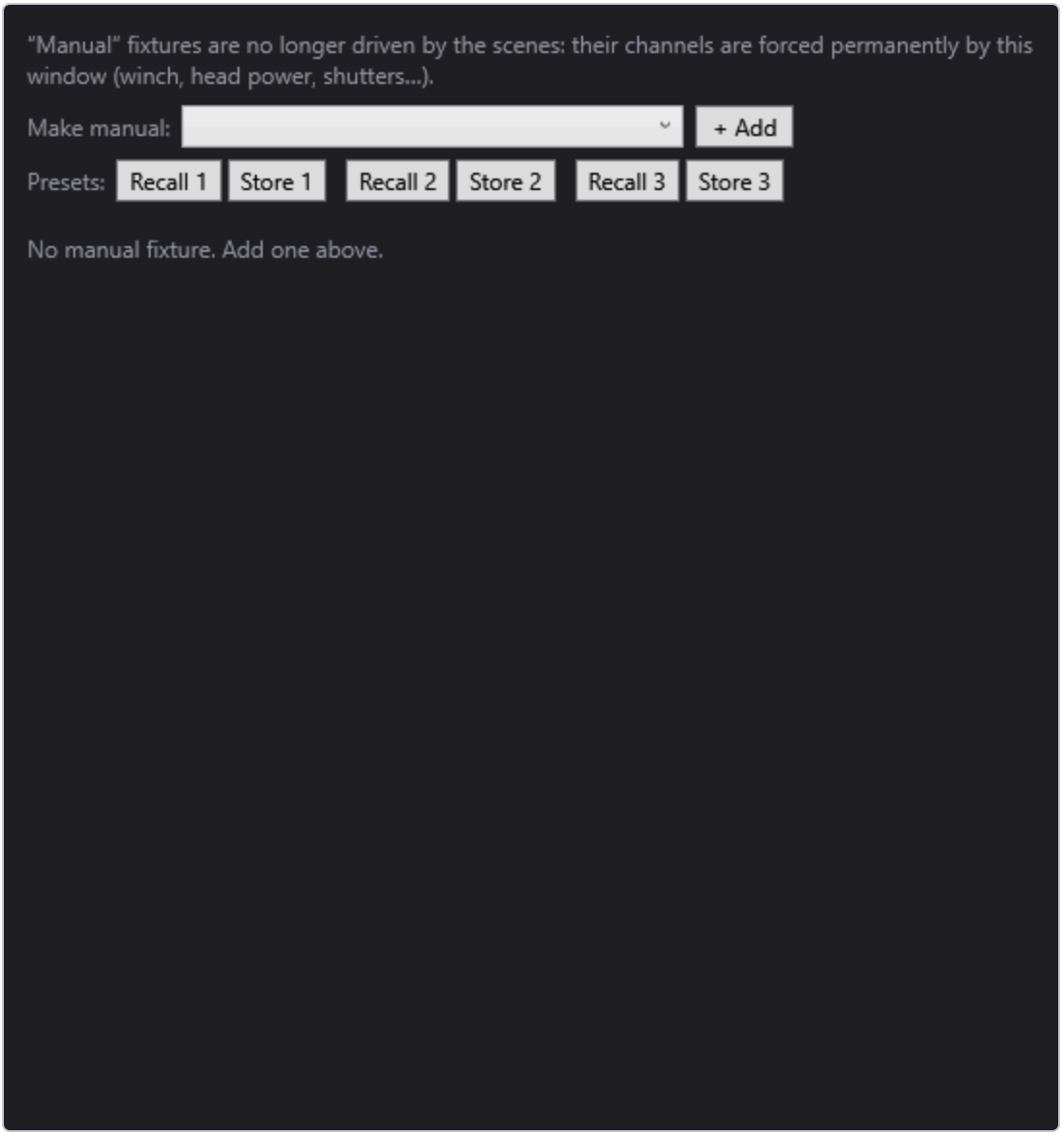
## See also

- Fixtures, addresses, positions
- DMX values and DMX faders
- Visualiser mode
- The standalone player
- Troubleshooting

# Manual controls

Some devices must not follow the cue list: a winch, the power supply of the moving heads, a relay, a mechanical shutter… The Manual controls window takes them out of the cue list and lets you drive them by hand. Their channels are forced permanently on the DMX output, whatever scene is playing, even during a general blackout.

**To open it:** menu **Venue > Manual controls…**, or the **Manual controls…** button of the cue window (show mode). The window does not block the main screen; if it is already open, the menu brings it to the front.




## The principle

- A **manual** fixture is no longer driven by the scenes: its channels take the values set in this window, at every DMX frame, on top of the cue list.
- These values go out **even during a general blackout**: a winch does not come back down and a power supply does not switch off because you blacked out.
- Two things still go on top of the manual controls: the follow spot on a tablet and the **DMX faders** (DMX menu), which override everything.
- Every setting is applied at once and **saved immediately** in the venue file: it is found again at the next launch.
- Manual fixtures are never switched on by the moods written automatically (first-show assistant).

## The window

At the top, a reminder: ""Manual" fixtures are no longer driven by the scenes: their channels are forced permanently by this window (winch, head power, shutters…)."

### Making a fixture manual

- **Make manual:** list of the venue's fixtures that are not manual yet, by identifier.
- **+ Add**: makes the chosen fixture manual. A card appears for it in the list.

Careful: a fixture that becomes manual for the first time starts with **all its channels at 0**, forced at once. Set it straight away.

### Presets

- **Presets:** three pairs of buttons **Recall 1** / **Store 1**, **Recall 2** / **Store 2**, **Recall 3** / **Store 3**.
- **Store N** saves the current values of **all** the manual fixtures in preset N (replacing its contents).
- **Recall N** puts these values back on the fixtures that are still manual. A fixture that became manual after the store is not touched. Recalling an empty preset does nothing.

Presets are saved with the venue.

### The manual fixture cards

Each manual fixture has its own card. The header gives its identifier, its universe and its address in the form "U1 @17" (U1 = the first universe, as in the DMX connections window), and its number of channels. The **Remove** button gives it back to the cue list (its settings are kept for the next time it becomes manual again). If there is no manual fixture: "No manual fixture. Add one above."

The card's contents depend on the device:

**A shutter** — a single-channel fixture whose identifier contains "shutter":

- **“Mode : AUTO”** / **“Mode : MANUEL”** (Mode: AUTO / Mode: MANUAL): switches between the two. In **AUTO**, the shutter follows the cue list (nothing is forced). In **MANUEL** (MANUAL), it is forced by the next button.
- **“OUVERT”** (OPEN) / **CLOSED**: toggles the shutter (255 = open, 0 = closed; a value of 128 or more is read as open). Inactive in AUTO mode.

**A relay or a power supply** — any other single-channel fixture:

- **“Sous tension (255)”** (Powered (255)) and **“Hors tension (0)”** (Unpowered (0)): set the channel to full or to zero.
- a 0–255 slider for an intermediate value, with the value shown on the right.

**A winch** — a fixture that has a channel named “Position” and a channel named “Speed”:

- two vertical faders, **“Position”** and **Speed**, from 0 to 255, with their value;
- **Zero**: resets **Position** to 0 (and the “Position fine” channel if there is one);
- **Stop**: stops the winch (sets **Speed** to 255, the winch's stop value);
- three **Recall N** / **Store N** rows specific to this winch: **Store N** keeps the current position (its value is shown at the end of the row), **Recall N** brings the **Position** fader back to that value. These positions are saved with the fixture;
- the winch's other channels (dimmer, colour…) as ordinary sliders.

**Any other device**: one 0–255 slider per channel, with the channel name on the left and the value on the right.

## Step by step

### Driving a winch by hand throughout the show

1. In the fixture library, check that the winch's definition has channels named “Position” and “Speed”.
2. **Venue > Manual controls…**.
3. In **Make manual:**, choose the winch, **+ Add**.
4. Set **Speed**, then raise **Position** to the height you want.
5. **Store 1** on the winch's first row. Do the same for other heights (**Store 2**, **Store 3**).
6. During the show, open the window from the cue window (**Manual controls…**) and use **Recall 1**, **Recall 2**… to send the winch to the stored heights.

### Keeping the moving heads' power on whatever happens

1. Patch the power relay as a one-channel fixture (in the venue, tick **“Invisible”** (Invisible) if it has no place on the stage).
2. **Venue > Manual controls…**, choose the relay, **+ Add**.
3. Click **Powered (255)**. The relay stays powered, even during a general blackout.

### Preparing two venue configurations

1. Make the devices concerned manual and set them for the first configuration (for example “rehearsal”). **Store 1**.
2. Set them for the second (“show”). **Store 2**.
3. Switch from one configuration to the other with **Recall 1** and **Recall 2**.

## Good to know

- **Manual = outside the cue list**: what the scenes say about this fixture no longer goes out (except a shutter in **AUTO** mode). To give it back to the cue list, **Remove**.
- **Difference from “hold”**: the venue's **If not mentioned in a scene: hold** setting lets the fixture follow the scenes, but makes it keep its value when a scene does not mention it, and protects it from the general blackout. A manual fixture, on the other hand, no longer follows the scenes at all. See Fixtures, addresses, positions.
- The **DMX faders** override the manual controls: a channel forced by a fader ignores the manual value until it is released.
- If you save the venue from **Fixtures, addresses, positions…** (or change venue) while this window is open, it updates by itself and its sliders keep working.
- Conversely, the **Fixtures, addresses, positions…** window works on a copy of the venue taken when it was opened: saving from it undoes the fixtures made manual and the settings made here in the meantime.

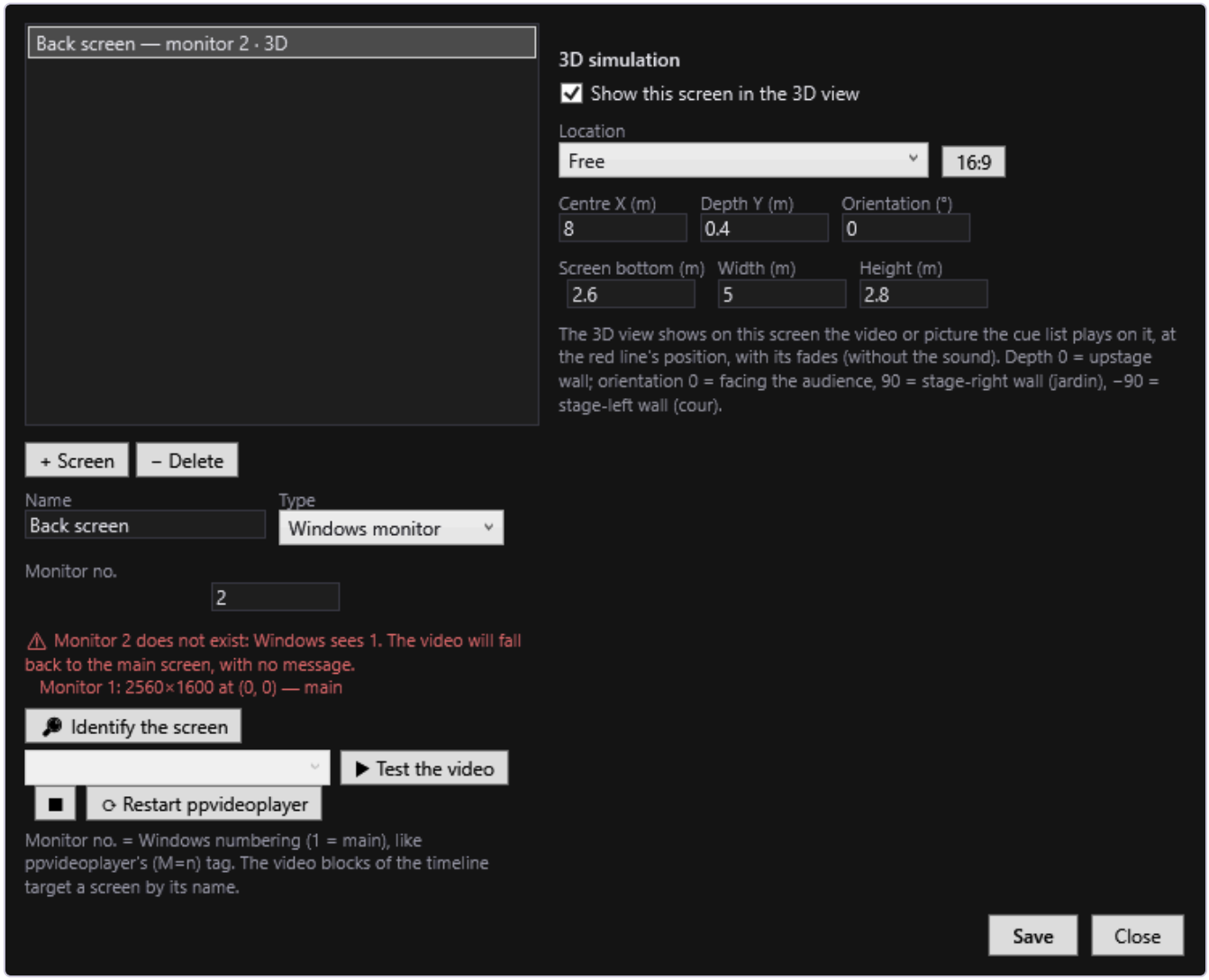
## See also

- Fixtures, addresses, positions
- DMX values and DMX faders
- Show mode and the cue window
- Follow spot on a tablet
- The fixture library

# Video screens

Screens are the surfaces on which the show plays its videos and pictures: a monitor or video projector plugged into the PC, or an IP video player (a box on the network). Each screen has a **name**: the timeline's video blocks target a screen by that name. A screen can also be placed in the venue so that the 3D simulation shows the video in its place.

**To open it:** menu **Venue > Screens...**. The window is titled “DreamShow (version) — Screens: (venue name)”. It does not block the main screen; if it is already open, the menu brings it to the front.




## Two kinds of screen

- **Windows monitor**: a screen plugged into the PC (monitor, video projector, LED wall through its graphics card). The video is played by the ppvideoplayer player installed with DreamShow,

which starts minimised.
- **IP video player**: a player box on the network (smallvp box). It connects to the show control PC and plays content **stored on itself**, filed by "slot" (A to P): you associate a file with each slot, then upload it to the box from this window.

## The list and the screen's properties (on the left)

- **The list**: one line per screen: "name — monitor N" or "name — IP: identifier", followed by "· 3D" if the screen is placed in the venue.
- **+ Screen**: adds a "Screen N" screen, of monitor type, on monitor N.
- **− Delete**: deletes the selected screen (without confirmation; nothing is saved before **Save**).
- **Name**: the screen's name, the one the timeline's video blocks choose. If you change it, blocks already placed that targeted the old name no longer find it.
- **Type**: **Windows monitor** or **IP video player**.
- **Monitor no.** (Windows monitor only): the monitor's number, in Windows numbering (1 = main), the same as ppvideoplayer's.

A note reminds you: "Monitor no. = Windows numbering (1 = main), like ppvideoplayer's (M=n) tag. The timeline's video blocks target a screen by its name."

### Tools for a Windows monitor

- **The list of monitors**: "Windows sees N monitor(s) — same numbering as ppvideoplayer:", then one line per monitor: "Monitor N: width×height at (x, y)", with "— main" for the main screen. If the chosen number does not exist, the text turns red: "⚠ Monitor N does not exist: Windows sees M. The video will fall back to the main screen, with no message."
- 🔎 **Identify the screen**: shows for 8 seconds, on this monitor, a full-screen card with its big number, "MONITEUR N" (MONITOR N), the screen's name, its resolution (and "main screen" where applicable). A click or a key closes it sooner. This is the sure way to know which physical screen carries which number.
- **The list of the show's videos** and **▶ Test the video**: plays the chosen video on this monitor through ppvideoplayer, that is, through the show's real path. The test stops by itself after 10 seconds. Careful: a video that starts on black will show nothing; use **Identify the screen** first. If the show has no video, the list is empty ("This show has no video: add one to test.").
- ■: stops the test.
- ⟳ **Restart ppvideoplayer**: restarts the player. ppvideoplayer only discovers monitors when it starts: do this after plugging in or enabling a screen.

## The IP video player (on the right)

This panel only appears for a screen of type **IP video player**.

- **ID or IP address**: designate the box by its identifier (set on the box) **or** by its IP address. Next to it, the live status: "● connected" or "○ not connected".
- A note reminds you that the box connects to the show control PC (port 5555) and plays local content by slot.
- **The list of connected boxes**: "Connected boxes: identifier (IP address), …" or "No box connected yet." It tells you what to type in **ID or IP address**.
- **+ Slot**: adds a slot (the first free letter from A to P).
- **Upload everything**: uploads all the slots, one after the other.

The videos the show already plays on this screen (video blocks that target its name) are added automatically to the list of slots, each on a free letter: you only have to check and upload. If there are none: "No video of the show targets this screen. Add a video block on the timeline (screen = this name), or "+ Slot" to prepare content."

Each slot has its own line:

- **Video**: the associated file (typed, or chosen with **File…**: videos .mp4, .mov, .avi, .mkv, .webm or images .png, .jpg, .jpeg, .bmp, .gif). A file taken from the show's media folder is saved by its relative name.
- **Slot**: the letter, from A to P.
- **Type**: **Video** or **Image**.
- **Upload**: sends the file to the box, then waits for its confirmation. The status is shown under the line: " ⏳ Upload in progress…", then ✓ (confirmed), ⚠ (sent but not confirmed in time) or ✕ (failed, with the reason: server unavailable, ID or IP missing, no file associated…). The window stays usable during a large transfer.
- ▶: test, plays this slot on the box.
- ■: stops playback on the box.
- ✕: removes this slot.

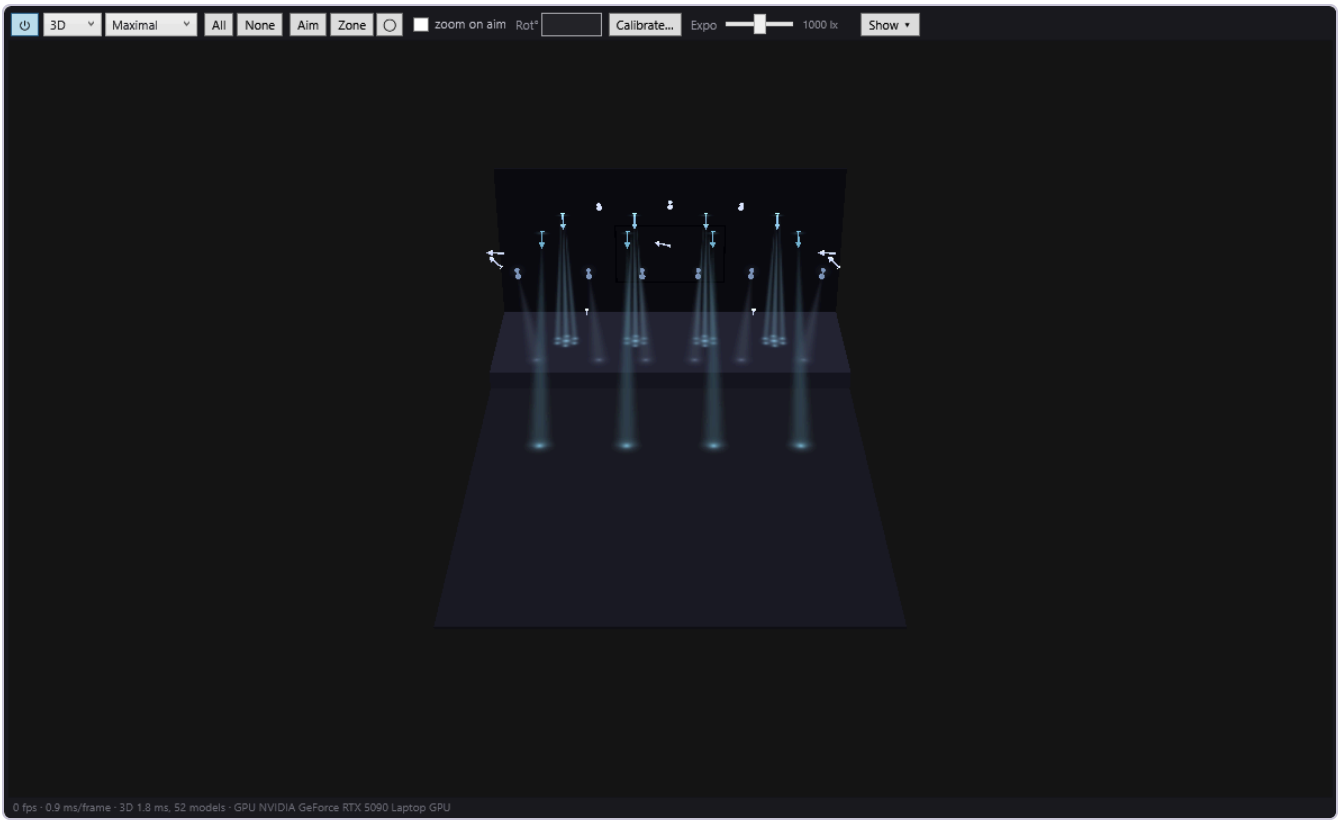
## The 3D simulation section (on the right)

It places the screen in the venue, so that the simulation shows it.

- **Show this screen in the 3D view**: when ticked, the screen is placed. On the first tick, it is put in the middle of the back wall, its bottom 1 m above the stage, as wide as half the venue (6 m at most) and in 16:9 format. When unticked, the screen is no longer drawn.
- **Location**: **Upstage wall (facing the audience)**, **Stage-right wall (jardin)**, **Stage-left wall (cour)**, **Back of the hall (facing the stage)** or **Free**. Choosing a wall sets the orientation and puts the screen against that wall (keeping its position along the wall when possible). **Free** is shown when the values match no wall.
- **16:9**: sets the height to width × 9/16.
- **Centre X (m)**: position of the screen's middle across the width, from the stage-right wall.
- **Depth Y (m)**: position in depth; 0 = the upstage wall.

- **Orientation (°)**: 0 = facing the audience, 90 = on the stage-right wall, −90 = on the stage-left wall, 180 = facing the stage.
- **Screen bottom (m)**: height of the lower edge, from the venue floor.
- **Width (m)** and **Height (m)**: size of the picture.

In the 3D view, the screen shows the video or picture the cue list plays on it, at the position of the timeline's red line, with its fades, without the sound. On the 2D plan, the screen is a thick line with a small mark on the side it faces and its name; the line lights up in blue when the screen is playing.




## Saving

- **Save**: applies the screens to the current venue and saves its file. Only the screens are concerned: the venue's other settings are not touched. As long as this has not been done, an asterisk “*” ends the window's title.
- **Close**: closes the window. If changes remain: “Screen changes not saved. Save?” (**Yes**, **No**, **Cancel**). A video test in progress is stopped.

## Step by step

### Declaring a video projector plugged into the PC

1. Plug in and switch on the video projector, set Windows to “Extend” mode.
2. **Venue > Screens...**, then **+ Screen**.

3. Give it a meaningful **Name** (for example “Back screen”), **Type** = **Windows monitor**.
4. Type a **Monitor no.**, then 🔎 **Identify the screen**: the card should appear on the video projector. If not, try another number.
5. If the video projector was plugged in after launch, click ⟳ **Restart ppvideoplayer**.
6. ▶ **Test the video** to check the whole chain.
7. **Save**. The timeline's video blocks can now target “Back screen”.

### Preparing an IP video player

1. Plug the box into the same network as the show control PC.
2. **Venue > Screens…**, **+ Screen**, **Type** = **IP video player**.
3. In **ID or IP address**, type the identifier or address shown in “Connected boxes”. The status should change to “● connected”.
4. Place your video blocks on the timeline with this screen: reopen the window, and their files appear in the slots.
5. **Upload everything**, then wait for the ✓.
6. Test a slot with ▶.
7. **Save**.

### Seeing the video in the 3D simulation

1. Select the screen, tick **Show this screen in the 3D view**.
2. Choose the **Location** (for example **Upstage wall (facing the audience)**), adjust **Width (m)**, **16:9**, **Screen bottom (m)**.
3. **Save**.
4. Switch the simulation to 3D and move the red line onto a video block: the picture appears on the screen.

## Good to know

- A screen declared on a monitor that does not exist plays **with no message** on the main screen: check the list of monitors and use **Identify the screen**.
- ppvideoplayer only sees monitors when it starts: after plugging in a screen, ⟳ **Restart ppvideoplayer**.
- An IP video player does not read a file by its name across the network: it plays what has been **uploaded** to its slot. After changing a video, upload it again.
- Renaming a screen does not update the video blocks already placed.
- The show's media are stored in its media folder (**File > Open the show's media folder**).

## See also

- Sound and video on the timeline

- 2D and 3D simulation
- Shows, files and projects
- Troubleshooting

# Switching, renaming, duplicating a venue

A **venue** describes a place: its dimensions, its fixtures and their addresses, its groups, its screens, its DMX connections, its calibrations. You can have several (your theatre, a touring venue, the office for tests) and switch from one to another. A venue can serve several shows.

**To get there: Venue** menu, last section: **Change venue**, **Rename the venue…**, **Duplicate the venue…**, **Delete the venue…**. Calibrations are imported and exported with **Import the calibrations (lyrescal.ini)…** and **Export the calibrations…**, in the same menu.

The active venue is recalled in the main screen's title, in the form “show @ venue”.

## The link between a show and its venue

- A show contains no DMX addresses: it designates fixtures (by their identifier) and groups (by their name), declared in the venue.
- Each time you save a show, it remembers the **name** of the active venue.
- When you open a show, DreamShow switches by itself to its venue if it is different from the active venue. If that venue no longer exists, the active venue is kept and the log says so: “The show references venue “…”, not found — current venue kept (…)”.

Consequence: if you change venue then save the show, it now belongs to the new venue.

## Changing venue

**Venue > Change venue** opens a submenu with one entry per venue (their name; the tooltip gives the file name). The active venue is ticked. Click another venue to activate it:

- the patch, the selection tree, the filters and the simulation are reloaded;
- the DMX output restarts on the new venue's connections;
- the log notes “Active venue: …”.

The open show is not modified: its scenes now target the fixtures and groups of the same name in the new venue. Whatever does not exist there does not light up.

### Creating a venue

At the bottom of the submenu, **New venue…** asks for the **Venue name:**. DreamShow creates an empty venue (12 × 18 × 6 m, one universe, the shared fixture library) and activates it. All that remains is to declare its fixtures: venue wizard or Fixtures, addresses, positions.

The file is named after the name (lower case, hyphens), with the extension `.venue.json`, in the data folder. If a file of that name already exists, creation is refused (“The file … already exists.”): choose another name.

## Rename the venue…

Renames the **active** venue. A window asks “New name (current: …):”.

- The new name is saved in the venue.
- All the shows in the data folder that referenced the old name are updated, as is the open show: automatic opening of the right venue keeps working. The log gives the number of shows updated.
- The venue's **file** name does not change.

## Duplicate the venue…

Creates a complete copy of the active venue (fixtures, groups, screens, calibrations, connections, manual presets…) under a new name. A window asks “Name of the copy of “…”:”. The copy is created in a new file and then **becomes the active venue**.

It is the right starting point for a neighbouring venue (same fixture inventory in another place) or for trying out changes without touching the original. As for a new venue, the name must not give a file that already exists.

## Delete the venue…

Deletes the **active** venue's file, after confirmation: “Permanently delete venue “…” (file)? Shows referencing it will say so when opened.”

- The last venue cannot be deleted (“The last venue of the project cannot be deleted.”).
- DreamShow first switches to another venue (the first in file-name order), then deletes the file.
- Deletion is permanent: the file is not put in the recycle bin. Remember to export the project beforehand (**File > Export the whole project (zip)…**) or to duplicate the venue if you are in doubt.

## Importing and exporting calibrations

Moving-head calibrations (see Moving-head calibration) are saved in the venue. These two commands save them in an .ini text file and reload them, for example after a change of venue or to take over the calibrations of an old installation.

- **Export the calibrations…**: saves an .ini file (`calibrations.ini` by default), with one section per calibrated moving head, named by the fixture's identifier. The log gives the number of

calibrations exported.
- **Import the calibrations (lyrescal.ini)…**: reads such a file back. Each section is attached to the fixture with the same identifier. A ppshow `lyrescal.ini` file (sections `[lyre1]` to `[lyre5]`) is accepted too: `[lyre1]` goes to the first fixture of the group named "Lyres" (French for moving heads), `[lyre2]` to the second, and so on. Only the sections marked as calibrated are taken over. Imported calibrations replace those of the fixtures concerned, and the venue is saved at once. If nothing is recognised: "No calibration recognised in this file."

## Step by step

### Preparing a tour in a second venue

1. Open the show (DreamShow activates its usual venue), then **Venue > Duplicate the venue…** and name the copy (for example "Tour — Village hall").
2. The copy is active and the show stays open: adjust the dimensions, positions and addresses in **Venue > Fixtures, addresses, positions…**, and the outputs in **Venue > DMX connections…**.
3. Check the show in the simulation, then save it **under another name** (**File > Save as…**): this version of the show belongs to the tour venue, the original stays linked to your venue. (If you reopened the original before this save, DreamShow would switch back to the usual venue.)

### Giving a show back to its original venue

1. **Venue > Change venue**, choose the original venue.
2. **File > Save**: the show remembers this venue again.

## Good to know

- Venues are `.venue.json` files in the data folder; they travel with **File > Export the whole project (zip)…**.
- Changing venue does not rewrite the show as long as you do not save it.
- The fixture library is shared by the venues: changing a definition changes every venue that uses it.
- Deleting a venue deletes no show: those that referenced it stay, and say so when opened.

## See also

- Fixtures, addresses, positions
- DMX connections
- Moving-head calibration
- Declaring your fixtures (venue wizard)
- Shows, files and projects

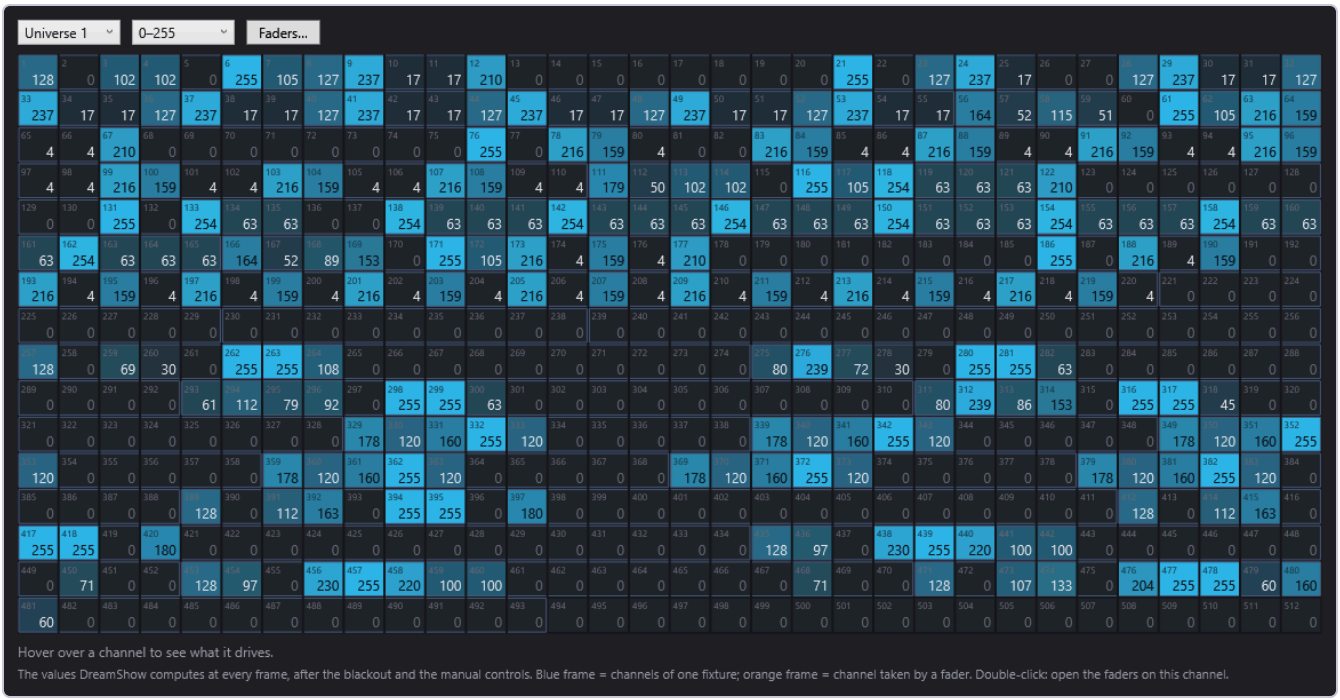
# DMX values and DMX faders

Two windows in the **DMX** menu for seeing and forcing what DreamShow sends to the fixtures. **DMX values…** shows live the 512 channels of a universe, as they go out. **DMX faders…** lets you take a channel by hand, over the cue list, to test a fixture or troubleshoot during a rehearsal.

**To open it:** menu **DMX > DMX values…** or **DMX > DMX faders…**. The faders also open from the DMX values window (**Faders…** button or double-click on a channel).

Reminder: a DMX **universe** has 512 channels. Each fixture occupies one or more consecutive channels starting at its address. A DMX value goes from 0 to 255.

## The DMX values window




The window shows the values **calculated by DreamShow at every frame**, that is after the general blackout, the manual controls and the faders: these are exactly the values that go out to the interfaces. It updates ten times a second.

### The top bar

- **Universe 1**, **Universe 2**…: chooses the universe displayed. The list contains all the universes used by the venue and its connections.
- **0–255** / **%**: unit of the values displayed in the grid. **0–255** (default) shows the raw DMX value; **%** shows it as a percentage (255 = 100 %). This choice only changes the display.

- **Faders…**: opens the DMX faders window on the universe displayed, from channel 1. To open it directly on a specific channel, double-click that channel in the grid instead.

### The grid

The 512 channels are laid out in 16 rows of 32 cells. In each cell:

- the small number at the top left is the channel's address (1 to 512);
- the large number at the bottom right is its value (0–255 or %);
- the cell fills with a colour that goes from dark grey (0) to bright cyan (255), to spot lit channels at a glance.

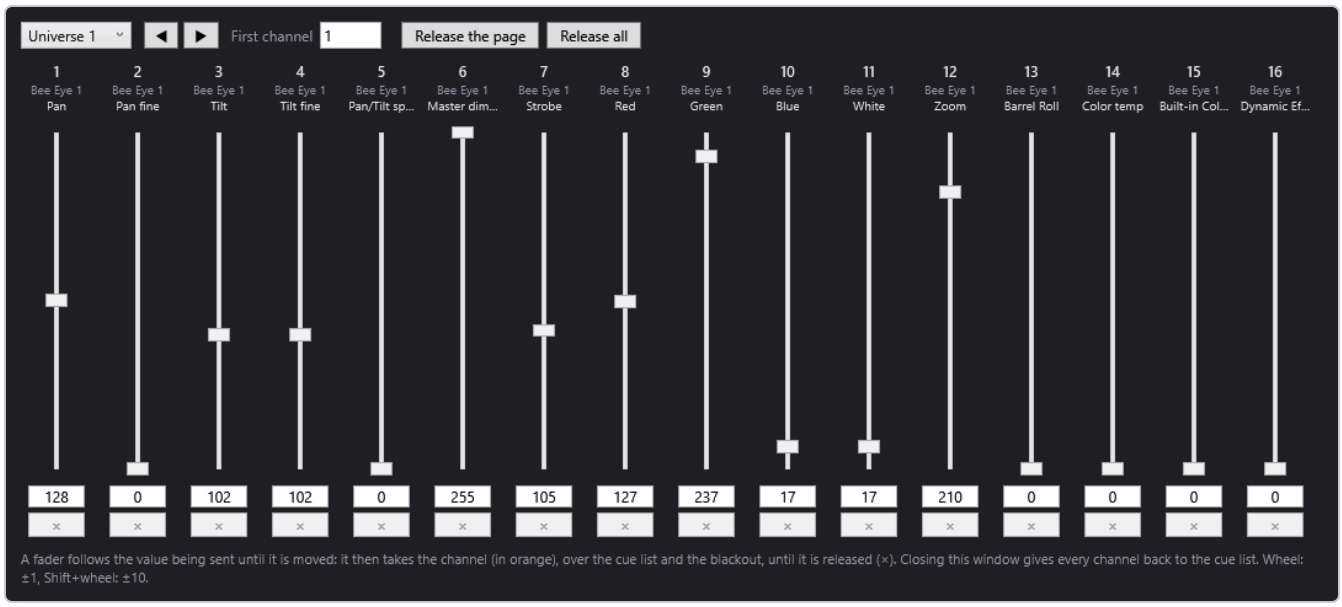
The frames:

- **Blue frame**: surrounds the channels of one fixture, according to the address and DMX mode declared in the venue. A channel with no frame is used by no fixture (“free channel”).
- **Orange frame**: the channel is taken by a fader (see below). Its value no longer comes from the cue list.
- **White frame**: the cell under the mouse pointer.

### Hovering and double-clicking

- **Hovering** over a cell displays at the bottom of the window what that channel drives, for example: “U1 · channel 12 = 255 (100 %) — Lyre 1 · Dimmer”. The name is that of the fixture in the venue, followed by the channel's name in its definition. If the channel is taken by a fader, the line ends with “· taken by a fader”. With no hover, the line says “Hover over a channel to see what it drives.”
- **Double-clicking** a cell opens the DMX faders window on the page that contains that channel (for example channels 33 to 48 for channel 37).

## The DMX faders window




The window presents a page of **16 faders**, one per consecutive channel.

## The principle: follow, then take

- A fader **follows** the value sent as long as you do not touch it: it rises and falls with the cue list.
- As soon as you **move** it (or type a value), it **takes** the channel: its frame and its number turn orange, and the chosen value is imposed at every frame, **over everything else**: the cue list, the general blackout, the manual controls and the follow spot on a tablet.
- The channel stays taken until you **release** it. It then goes back to the cue list.

## The top bar

- **Universe 1**, **Universe 2**…: the universe whose channels are displayed. Changing universe releases nothing: channels already taken on the other universe stay taken.
- ◀ (previous page) and ▶ (next page): goes back or forward by 16 channels.
- **First channel**: type the number of the first channel displayed (1 to 497), then Enter or click elsewhere. The page then starts exactly at that channel.
- **Release the page**: gives the 16 channels displayed back to the cue list.
- **Release all**: gives all the channels taken back to the cue list, on all universes.

## Each fader

From top to bottom:

- **the channel number** (orange when taken);
- **the name of the fixture** that uses this channel, or "—" for a free channel;
- **the name of the channel** in the fixture's definition (Dimmer, Pan, Red…). A tooltip reminds you "Channel N — fixture · channel";

- **the vertical slider**, from 0 (bottom) to 255 (top). A click on the track places the slider directly at that spot;
- **the value box**: type a value from 0 to 255 then Enter (or leave the box) to take the channel at that value. Any % sign is ignored: “50%” gives DMX value 50, not 50 %.
- **×** (Release: the channel goes back to the cue list): active only when the channel is taken.

The mouse **wheel** over a slider changes the value by 1; **Shift + wheel** changes it by 10. Like any movement of the slider, the wheel takes the channel.

### Closing the window

**Closing the DMX faders window gives all the channels back to the cue list**, on all universes. This way, no forgotten setting stays imposed during a performance. The log then notes “Faders: every channel given back to the cue list”.

Exception: when you double-click another channel in the DMX values window while the faders are already open, the window is rebuilt on that channel **keeping** the channels already taken.

## Step by step

### Checking that a fixture is receiving its values

1. Open **DMX > DMX values...**.
2. Choose the fixture's universe.
3. Find its blue frame and hover over its cells: the bottom line gives the name of each channel.
4. Start the scene you want (or click **GO**): the values should move in the grid.
5. If the values move but the fixture does not react, the problem is between the PC and the fixture: connections, the fixture's address or mode, cable. See DMX connections and Troubleshooting.

### Testing a fixture channel by channel

1. In **DMX values...**, double-click the fixture's first channel. The faders open on its page.
2. Raise the intensity channel's fader (Dimmer): the fixture should light up. If the fixture has a shutter, it may need opening too.
3. Move the other channels one by one (colour, Pan, Tilt...) and watch the fixture.
4. Click **Release the page**, or close the window, to give everything back to the cue list.

### Holding a channel at a fixed value during a rehearsal

1. Open **DMX > DMX faders...**, choose the universe, type the channel in **First channel**.
2. Type the value in the box under the fader, then Enter.
3. Leave the window open as long as the value must be held. Click **×** to release it.

## Good to know

- The DMX values window shows the calculated values even when nothing goes out: **LIVE** button off in the editor, or interface unplugged. Values that move only prove that DreamShow is calculating the cue list.
- In Visualiser mode, the DMX values window shows the DMX **received** from the network, not DreamShow's. The faders also follow these received values, but taking a channel then has no effect on the venue, since DreamShow's output is cut.
- Turning off the simulation rendering (the ⏻ button of the simulation panel) stops neither the DMX output nor the DMX values window. See 2D and 3D simulation.
- The faders override the general blackout: a channel taken stays lit even during a **BLACKOUT**. Remember to release everything before a show.
- The same channels can also be taken from a MIDI controller, the web remote or OSC. These takes show in orange in both windows, and closing the DMX faders window releases them too.
- Channels taken are not saved: they disappear when DreamShow is closed.
- For devices that must stay set by hand permanently (winch, power, relays), use the manual controls instead, which are saved with the venue.

## See also

- DMX connections
- Manual controls
- Fixtures, addresses, positions
- Visualiser mode
- MIDI controllers
- The web remote
- The OSC remote
- Troubleshooting

# Visualiser mode

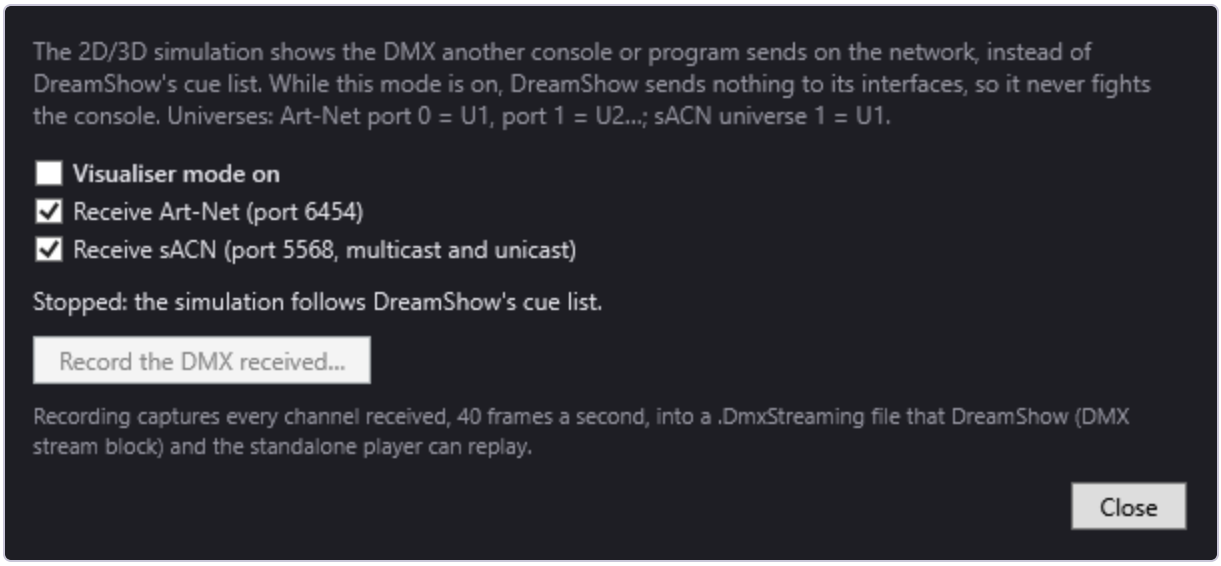
Visualiser mode turns DreamShow into a monitoring screen for other software or another lighting desk: the 2D/3D simulation shows the DMX that device sends over the network (Art-Net or sACN), instead of DreamShow's cue list. It also lets you record this received DMX into a file that DreamShow and the standalone player can replay.

Use it to preview in 3D a show programmed elsewhere, to check what a console is sending, or to capture an existing show to replay it later.

**To open it:** menu **DMX > Visualiser mode (Art-Net / sACN in)…**.

Two network protocols are recognised:

- **Art-Net**: DMX carried over the Ethernet network, very widespread.
- **sACN** (E1.31 standard): another way of carrying DMX over a network, often by multicast (one transmission received by all subscribed devices).




## What happens when the mode is active

- The 2D/3D simulation and the **DMX values** window display the **received** DMX.
- **DreamShow no longer sends anything to its own interfaces** (Art-Net, USB, uDMX), so as never to fight with the console driving the venue. This is also true in show mode: visualiser mode takes precedence.
- DreamShow's title bar shows it with “[visualiser — output off]”.
- The log notes “Visualiser mode: the simulation shows the DMX received from the network, DreamShow's output is off”, then “Visualiser mode stopped: DreamShow drives its output

again" when it stops.

## The window

- **Visualiser mode on**: tick box that starts or stops the mode. If reception is impossible (port already taken by another program, for example), a message explains why and the box unticks itself.
- **Receive Art-Net (port 6454)**: listens for Art-Net. Ticked by default.
- **Receive sACN (port 5568, multicast and unicast)**: listens for sACN, whether it is sent by multicast or directly to this PC's address. Ticked by default.

At least one of the two protocols is needed; otherwise the message "Tick Art-Net, sACN or both." is displayed. Changing a protocol while the mode is active restarts reception at once. These two choices are remembered.

### Universe correspondence

The window's help text reminds you: **Art-Net port 0 = U1, port 1 = U2…**; **sACN universe 1 = U1**. In other words, Art-Net counts from 0 and DreamShow from 1, whereas sACN and DreamShow both count from 1.

Only the current venue's universes are displayed. Packets for other universes are counted as ignored.

### The status line

It updates twice a second:

- "Stopped: the simulation follows DreamShow's cue list.": the mode is stopped.
- "Waiting for DMX… Check that the console sends to this PC (or by broadcast / multicast).": the mode is active, but nothing is received.
- "Received: U1, U2 · from 192.168.1.20 · 44 packets/s": the universes that received data in the last 3 seconds, the address of the devices sending them, and the number of packets per second. If packets target universes absent from the venue, the line adds " · N packets outside the venue's universes ignored".

### Recording

- **Record the DMX received…**: active only when the mode is active. It asks where to save the file; the suggested folder is the show's media folder, and the suggested name is of the type `capture-20261005-2130.DmxStreaming`.
- During recording, the button becomes ■ **Stop recording (mm:ss)** with the length recorded. Click it to finish: a message gives the file's path.

The recording captures **all the channels received, 40 frames per second**, in a **.DmxStreaming** file. It records the universes from U1 up to the highest universe receiving data at the moment you start recording.

- **Close**: closes the window. Visualiser mode (and a recording in progress) carries on; reopen the window to stop it.

## Step by step

### Seeing another console's show in 3D

1. Connect the PC to the same network as the console (or the lighting software).
2. In the console, set the Art-Net or sACN output to this PC's address, or to broadcast / multicast.
3. Check that DreamShow's venue declares the same fixtures, at the same universes and addresses (**Venue > Fixtures, addresses, positions…**).
4. Open **DMX > Visualiser mode (Art-Net / sACN in)…** and tick **Visualiser mode on**.
5. The status line should show “Received: …”. Switch the simulation to 3D to see the beams.

### Recording a show to replay it in DreamShow

1. Enable visualiser mode as above and check that all the expected universes appear in “Received: …”.
2. Click **Record the DMX received…** and keep the show's media folder.
3. Play the show on the console.
4. Click ■ **Stop recording**.
5. Untick **Visualiser mode on**: DreamShow drives its output again.
6. Saved in the show's media folder, the file appears in the **Media** list: drag it into a scene's timeline to create a DMX recording block (see The scene timeline). It can also be used by the standalone player.

## Good to know

- **Visualiser mode is never reactivated at the next launch**: a show cannot start with its output cut by a forgotten setting. Only the Art-Net / sACN choices are remembered.
- As long as the mode is active, nothing goes out of DreamShow, even in show mode and even with the **LIVE** button on. Remember to deactivate it before taking control of the venue again.
- If a console runs on the same PC, it keeps its own Art-Net port: DreamShow opens reception in shared mode.
- When the console stops transmitting, the simulation keeps the last values received.
- On first use, the Windows firewall may ask to allow DreamShow on the network: accept, otherwise nothing will be received.

- A universe that starts transmitting after the recording has begun, and whose number is higher than all the universes already active, is not recorded. Wait until the whole console is transmitting before starting the recording.

## See also

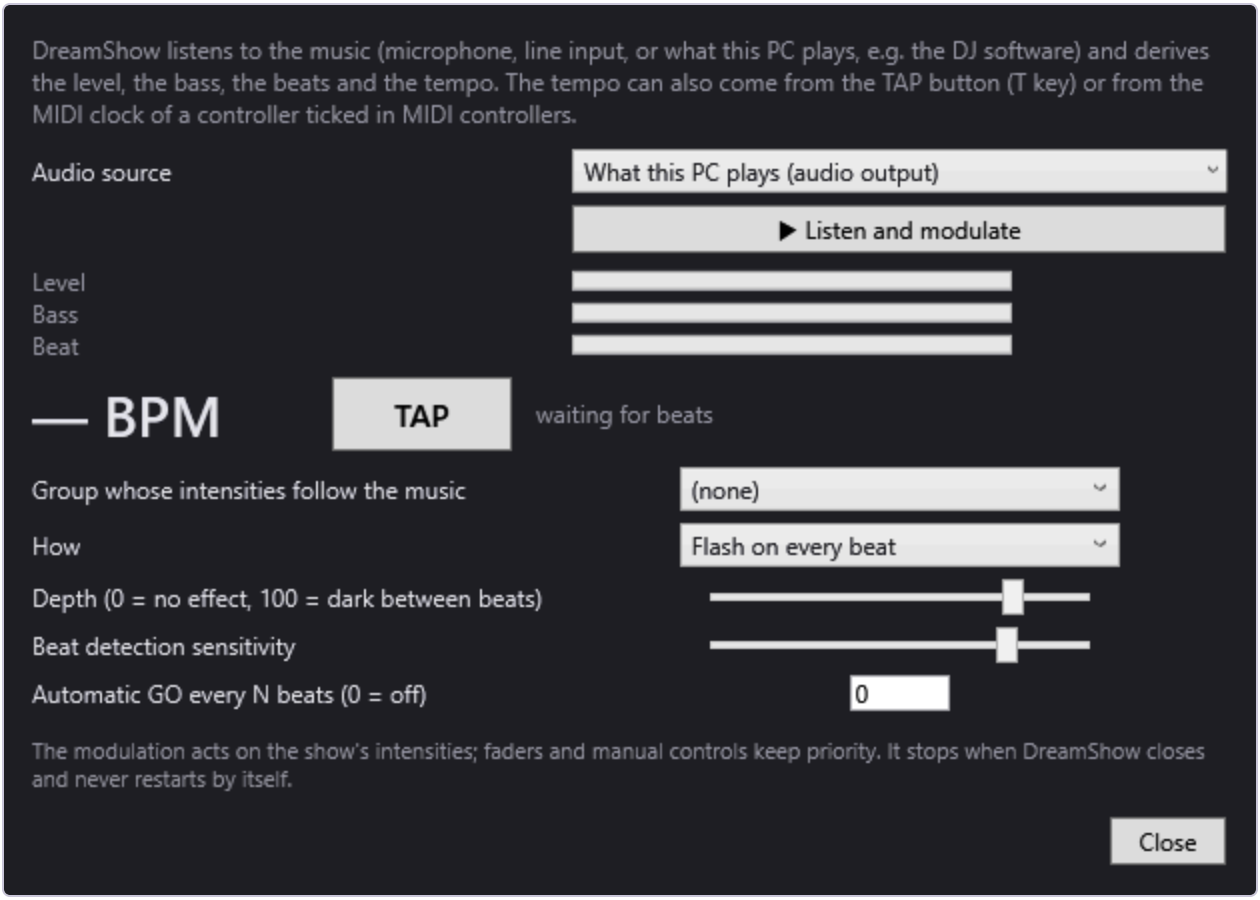
- 2D and 3D simulation
- DMX values and DMX faders
- DMX connections
- The standalone player
- DMX stream by an AI

# Sound-to-light and tempo

DreamShow can listen to music live (a microphone, a line input, or what the PC is playing, for example DJ software) and use it in two ways: vary the intensities of a group of fixtures to the rhythm of the sound, and provide a **tempo** (in BPM, beats per minute) that can fire a GO every N beats. The tempo can also be tapped by hand (TAP) or come from the MIDI clock of a desk or a program.

Useful for a dance party, a concert without a written cue list, or to keep a cue list locked to the rhythm.

**To open it:** menu **DMX > Sound-to-light and tempo...**.




## The audio source and listening

- **Audio source**: what DreamShow listens to.
  - **What this PC plays (audio output)** (default): the sound coming out of the PC, whatever program is playing it.
  - The other lines are Windows' sound inputs (microphone, a sound card's line input...), with their Windows name. Changing the source while listening immediately restarts

listening on the new source.

- ▶ **Listen and modulate**: starts listening and modulating the lights. The button becomes ■ **Stop listening**. If the source cannot be opened, a message “Audio input unavailable: …” explains why.

While listening, DreamShow's title bar displays “[sound-to-light]”.

## The indicators

- **Level**: the overall volume, relative to the loudest passage heard recently (the gauge adapts to the volume of the music).
- **Bass**: the energy of the low frequencies (below about 150 Hz), also relative.
- **Beat**: a flash on each beat, which falls back at once.
- **The tempo**, in large type: “124 BPM”, or “— BPM” as long as no tempo is known.
- **TAP**: tap this button to the rhythm of the music to give the tempo. The **T** key does the same when the window has focus (unless the cursor is in a text box).
- On the right, the origin of the tempo: **tapped tempo**, **MIDI clock**, **detected in the music** or **waiting for beats**.

### Where the tempo comes from

- **Detection in the music**: DreamShow picks out the strong hits in the bass (kick drum, bass) and deduces the tempo. The result is brought back between 70 and 180 BPM: a tempo detected at half or double is corrected. A few hits are needed to get a tempo.
- **TAP**: the tempo is the average of the intervals between the last taps (up to 8). A pause of more than 2.5 seconds restarts the count.
- **MIDI clock**: a mixing desk, DJ software or a sequencer sends 24 pulses per beat. The device must be ticked and applied in **DMX > MIDI controllers…** (see MIDI controllers). A MIDI “Start” message resets the beat count to zero. As long as the MIDI clock is running, it takes priority over detection.

The most recent source wins. Between two real beats, DreamShow keeps beating time at the last known tempo.

TAP and the MIDI clock work even without starting listening: the tempo is then displayed in the window and on the web remote.

## What the music does to the lights

- **Group whose intensities follow the music**: the group of fixtures that is modulated (the venue's groups). **(none)**: no light is modulated, only the tempo is used.
- **How**:
  - **None (tempo only)**: no modulation; the tempo and the automatic GO work.

    - **Follow the level**: the intensities follow the overall volume.
    - **Follow the bass**: the intensities follow the low frequencies.
    - **Flash on every beat** (default): the intensities are full on each beat and fall back just after.
- **Depth (0 = no effect, 100 = dark between beats)**: the strength of the modulation, from 0 to 100 (80 by default). At 0, nothing changes; at 100, the fixtures drop down to blackout when the music (or the flash) is at its lowest. In between, they only drop by part of the way.
- **Beat detection sensitivity**: towards the right, DreamShow detects beats more easily (lower threshold); towards the left, it only keeps very pronounced hits. If detection misses beats, push towards the right; if it invents some, push towards the left.

The modulation only **reduces** the intensities the cue list already gives the group: a fixture that is off in the scene stays off. It acts on the group's intensity channels only (not on colours or positions). The DMX faders and the manual controls keep priority, and so does the general blackout.

## The automatic GO

- **Automatic GO every N beats (0 = off)**: type a number of beats (for example 32) for a GO to go off every N beats. 0 (default) turns it off.

This automatic GO only works while listening (▶ **Listen and modulate** started), but the beats counted can come from any source (detection, TAP, MIDI clock). It follows the same rules as a GO given by hand: in Show mode, it only goes through if the GO is possible (scene looping, or scene timeline finished), and it does nothing on the last scene. A “Start” from the MIDI clock resets the beat count to zero.

- **Close**: closes the window. Listening and modulation carry on.

## Step by step

### Making PARs pulse to the DJ's music

1. The DJ software plays on this PC: choose **What this PC plays (audio output)** in **Audio source**.
2. In **Group whose intensities follow the music**, choose the PAR group.
3. In **How**, choose **Flash on every beat**.
4. Set **Depth** to about 80.
5. Click ▶ **Listen and modulate**. The **Beat** gauge should blink with the music and the tempo be displayed.
6. Make sure the current scene does light this group: the modulation only lowers what the scene gives.

### Changing scene every 8 bars

1. Prepare looping scenes (a looping scene accepts GO at any time).
2. In **Automatic GO every N beats (0 = off)**, type 32 (8 bars of 4 beats).
3. Start listening, or give the tempo with **TAP** then start listening.
4. Open Show mode: the cue list advances every 32 beats.

## Good to know

- **Listening never restarts by itself** the next time DreamShow is launched: a show cannot begin dimmed by a forgotten setting. The settings (source, group, mode, depth, sensitivity, automatic GO) are, however, remembered.
- With a microphone, place it near the bass source and away from the audience; a noisy room distorts detection.
- Detection does not accept more than one beat every 0.28 seconds or so.
- TAP also exists on a MIDI controller (**Tap tempo** action), on the web remote (**TAP** button) and in OSC ( `/dreamshow/tap` ).

## See also

- MIDI controllers
- Show mode and the cue window
- The web remote
- The OSC remote
- Dynamic effects

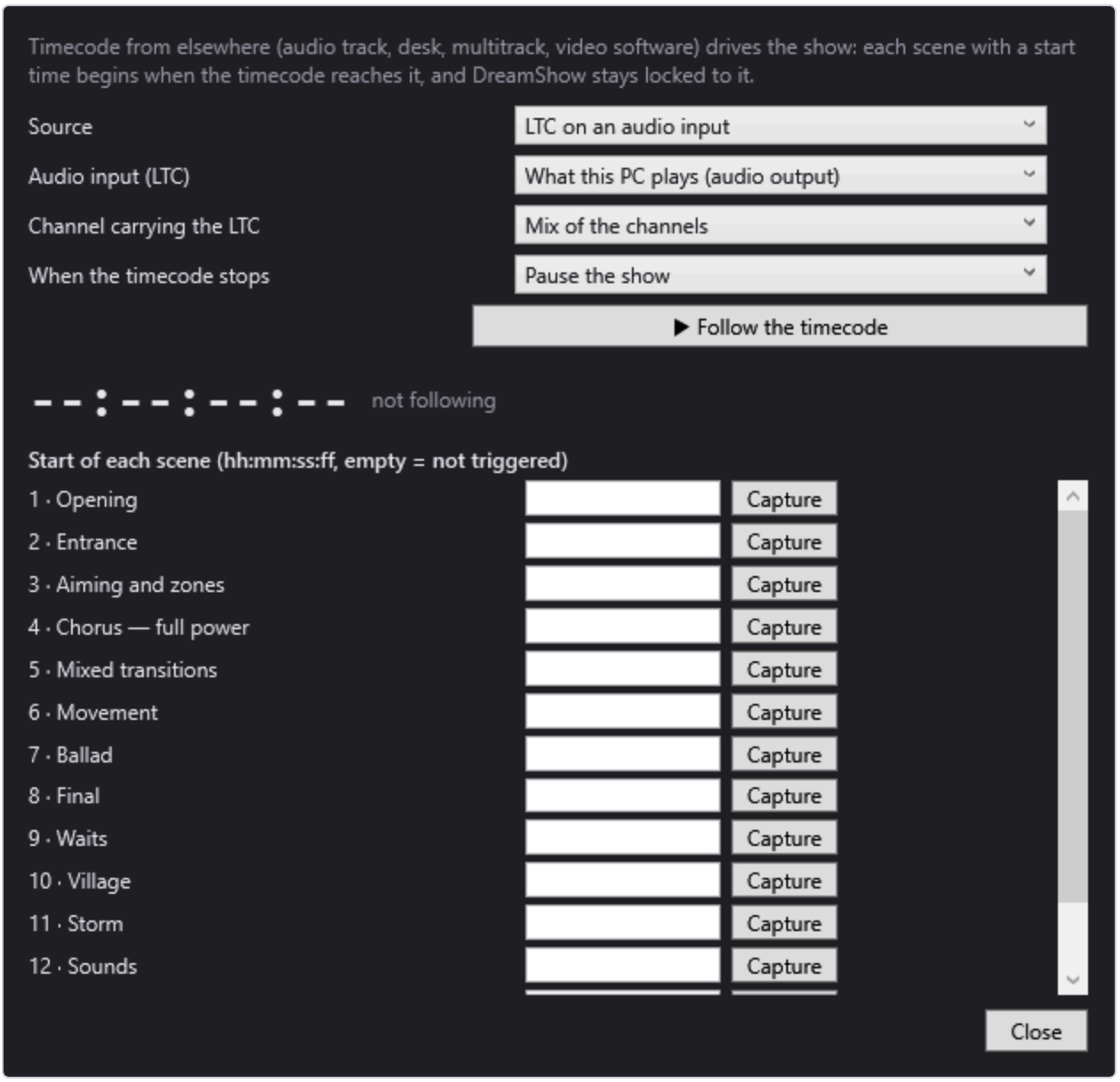
# Timecode (LTC / MTC)

**Timecode** is a time (hours:minutes:seconds:frames) sent continuously by another device: an audio track, a mixing desk, a multitrack recorder, video or music software. DreamShow can follow it: each scene that has a start time starts when the timecode reaches it, and DreamShow stays locked to it. It is the solution when the light must land exactly with a soundtrack or a video played by another system.

Two forms of timecode are recognised:

- **LTC** (linear timecode): an audio signal, carried on an audio track or an audio cable, arriving at a sound input of the PC.
- **MTC** (MIDI timecode): the same time carried as MIDI messages, arriving through a controller or a MIDI interface.

**To open it:** menu **DMX > Timecode (LTC / MTC)…**.




## The source

- **Source**:
  - **LTC on an audio input** (default).
  - **MTC from the MIDI controllers**: the MTC arrives through the MIDI devices ticked and applied in **DMX > MIDI controllers...** (see MIDI controllers).
- **Audio input (LTC)**: the sound input that receives the LTC. The first line, **What this PC plays (audio output)**, listens to the sound coming out of the PC; the following ones are Windows' inputs (line input, microphone...). Setting active only for LTC.
- **Channel carrying the LTC**: **Mix of the channels** (default), **Left channel** or **Right channel**. Choose the channel that carries the LTC when the other carries music, for example a stereo track with the music on the left and the timecode on the right. Active only for LTC.
- **When the timecode stops**:

    - **Pause the show** (default): when the timecode stops arriving, the cue list freezes where it is.
    - **Let the show carry on**: the cue list keeps playing by itself. The timecode is considered stopped when no frame has arrived for half a second.

Any change of source, input or channel immediately restarts reception if following is running. These settings are remembered.

## Following the timecode

- ▶ **Follow the timecode**: starts following. The button becomes ■ **Stop following the timecode**. If the sound input cannot be opened, a message “Audio input unavailable: …” explains why.
- **The large display**: the timecode received, in the format hh:mm:ss:ff (ff = frames). “--:--:--:--” as long as nothing has been received; when the timecode stops, the last value received stays displayed.
- **The status line** next to it:
    - **not following**;
    - **no timecode received** followed by the source, when following is running but nothing arrives;
    - during reception: the frame rate detected (for example “25 fps”), the source, and the scene the timecode points to (“scene 3”).

While following, DreamShow's title bar displays “[timecode]”.

### What DreamShow does while following

About 25 times a second, DreamShow looks for **the last scene whose start time has been reached** by the timecode, and calculates where we are in that scene. Then:

- if the cue list is not on that scene, it jumps to it, at the right position in its timeline, and makes it play;
- if it is on the right scene but out by more than a quarter of a second, or paused, it resets it and restarts it;
- once the end of a scene's timeline is reached (a scene that does not loop), it lets it rest on its end without restarting it.

Practical consequences:

- before the start time of the first scene, nothing happens;
- if the timecode goes backwards (rewind), DreamShow goes back to the corresponding scene;
- a scene without a start time is never chosen by the timecode;
- while the timecode is running, a GO or a click on another scene is corrected at once: the timecode decides.

Jumps are made without a fade, straight to the right position.

## The start of each scene

Under the heading **Start of each scene (hh:mm:ss:ff, empty = not triggered)**, each scene of the show has a line: its number and name, a start-time box, and a **Capture** button.

- **The box**: type the start time, for example `00:01:30:00`, then leave the box. The separators `:`, `;` and `.` are accepted; four numbers are required (minutes and seconds up to 59, frames up to 29). An invalid entry is cancelled and the old value comes back. An empty box = the scene is not triggered by the timecode.
- **Capture** (Take the timecode being received now as this scene's start): copies the timecode that is arriving, or failing that the last one received, into the box.

The start times are part of the show: they mark the show as modified, and it has to be saved (**File > Save**, Ctrl+S).

- **Close**: closes the window. Following carries on.

## Step by step

### Locking a show to a soundtrack that carries LTC

1. Connect the output that carries the LTC to a line input of the PC.
2. Open **DMX > Timecode (LTC / MTC)...**, choose **LTC on an audio input**, the line input, and the channel that carries the LTC.
3. Click ▶ **Follow the timecode** and start the tape: the display should scroll and the frame rate be shown.
4. For each scene, at the right moment of the tape, click **Capture**; or type the start times noted from the cue sheet.
5. Save the show (Ctrl+S).
6. Rewind and restart the tape: the scenes all start by themselves at their time.

### Following music software over MTC

1. In **DMX > MIDI controllers...**, tick the MIDI interface that receives the MTC, then **Apply**.
2. In **DMX > Timecode (LTC / MTC)...**, choose **MTC from the MIDI controllers**.
3. Click ▶ **Follow the timecode**, then start playback in the software.
4. Fill in the scenes' start times as above.

## Good to know

- **Following never restarts by itself** when DreamShow is launched. Remember to click ▶ **Follow the timecode** before the show. The source settings, however, are remembered.
- The frame rate (24, 25, 29.97 or 30 frames per second) is detected automatically.
- Frames typed by hand in a start box are counted at 25 frames per second, whatever the frame rate received. With a 30 fps timecode, prefer **Capture**, or times without frames ( `…:00` ).
- Following works in the editor as in Show mode. In the editor, nothing goes out to the fixtures if the **LIVE** button is off.
- For LTC, set a sufficient input level: a signal that is too weak is not decoded.

## See also

- MIDI controllers
- Show mode and the cue window
- The scene timeline
- Waits (key, time, date, sun)

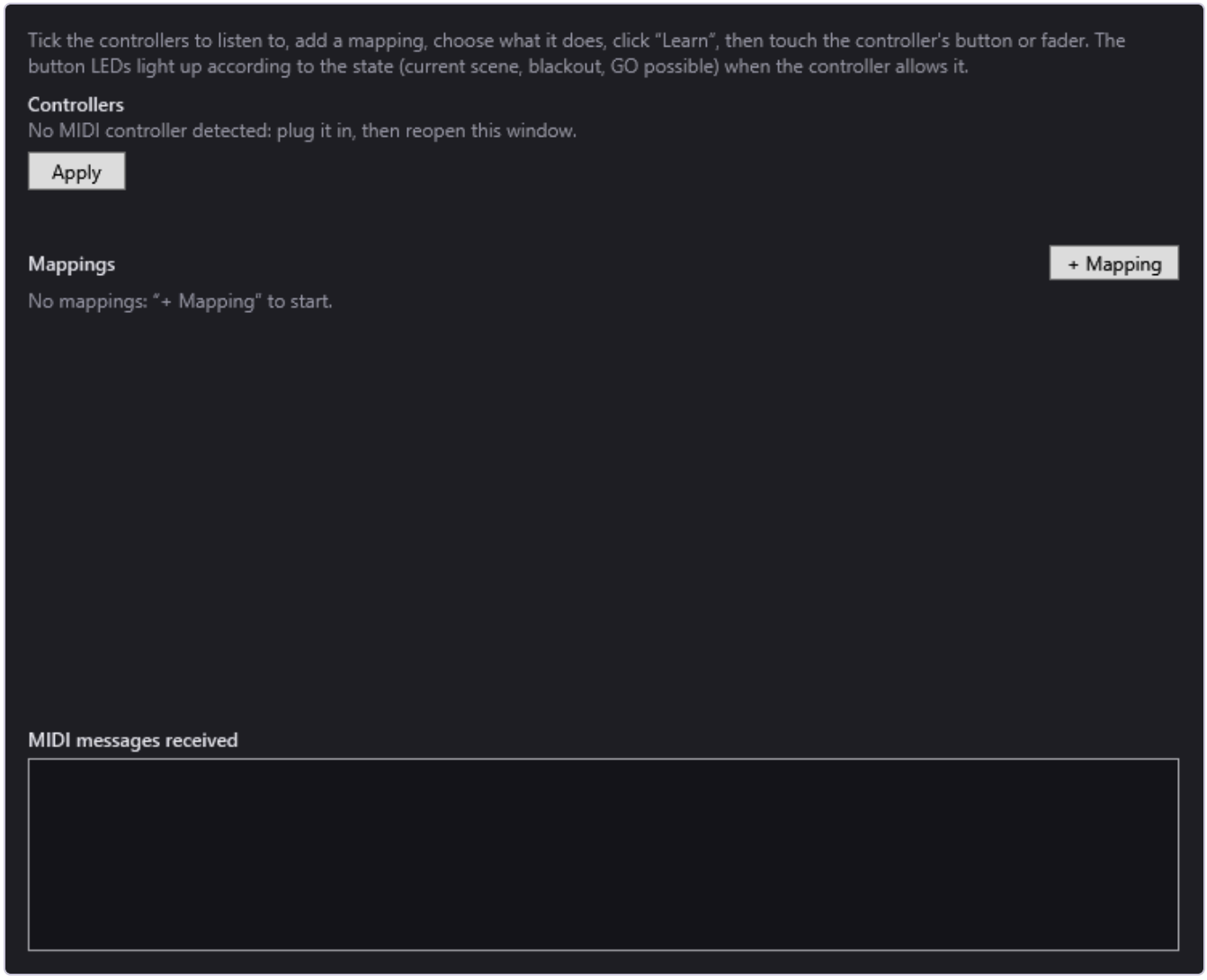
# MIDI controllers

A MIDI controller (a surface with pads, buttons or faders, plugged in by USB) can drive DreamShow: Go, previous scene, jump to a scene, blackout, pause, DMX faders… You **learn** each button or fader by touching it. When the controller allows it, its LEDs light up according to the state of the show (current scene, blackout, Go possible).

The controllers ticked here also serve as the input for the **MIDI clock** (Sound-to-light and tempo) and for **MTC** (Timecode).

**To open it:** menu **DMX > MIDI controllers…**.

No special driver is needed: USB controllers recognised by Windows as MIDI devices appear directly.




## The controllers

- **Controllers**: the list of detected MIDI inputs, each with a tick box. Tick those that DreamShow must listen to. A controller ticked in a previous session but absent today appears with the note "(unplugged)".
- If nothing is detected: "No MIDI controller detected: plug it in, then reopen this window."
- **Apply**: opens the ticked controllers and saves this choice. **The boxes only take effect after Apply.**
- The status line under the button says "MIDI: name1, name2" for the open controllers, or "No MIDI controller open".

The ticked controllers are reopened automatically when DreamShow starts. A controller that is ticked but absent is flagged in the messages by "Controller missing: name".

## Mappings

A **mapping** links a button or a fader of the controller to an action in DreamShow.

- **+ Mapping**: adds a line (GO action by default). With no line at all: "No mappings: "+ Mapping" to start."

Each line contains, from left to right:

- **The action** (list): see the table below.
- **The argument**: a small box, active only for actions that need one. Tooltip: "Scene: its number. Fader: universe.channel, for example 1.25." For a universe and a channel, you can also write `1/25`.
- **The learned control**: "—" as long as nothing is learned; then for example "Note 36 · ch 1 · controller name" for a pad or a button that sends notes, or "CC 7 · ch 1 · controller name" for a fader or a button that sends "Control Change" messages.
- **Learn**: puts the line on standby; the text becomes "touch the control…" in orange and the button becomes **Cancel**. Then touch the controller's button or fader: it is associated with the line at once and saved.
- ✕ (Delete this mapping).

A mapping only responds to the controller, MIDI channel and note (or CC number) learned.

### The actions

| Action | Argument | What it does |
|---|---|---|
| **GO** | — | Go, with the same rules as the **GO** button (see below). |
| **Previous scene** | — | Goes back to the previous scene, with no fade. |
| **Go to scene no.** | scene number | Jumps to that scene (1 = first). |

| | | |
|---|---|---|
| **Blackout** | — | General blackout. A button that sends notes toggles the blackout at each press; a CC control turns it on from the middle of its travel (64) and off below that. |
| **Pause** | — | Pauses the cue list. |
| **Play** | — | Restarts the cue list. |
| **Ready (reset)** | — | Goes back to the "Ready" state: blackout, media stopped, the next Go starts scene 1. |
| **LIVE output** | — | Turns the editor's **LIVE** button on or off (same behaviour as **Blackout** between note and CC). |
| **DMX fader (universe.channel)** | universe.channel | Takes this DMX channel, like a fader of **DMX > DMX faders...**. A fader (CC from 0 to 127) gives a value from 0 to 255; a pad (note) gives its velocity converted the same way, and 0 on release. |
| **Give the channels back to the cue list** | universe.channel, or empty | Releases this channel; with no argument, releases all the channels taken. |
| **Tap tempo** | — | One tempo tap, like the **TAP** button of Sound-to-light and tempo. |

For button-type actions (GO, Previous scene, Go to scene, Pause, Play, Ready, Give the channels back, Tap tempo), only the press counts: a note pressed down, or a CC at 64 or more. The release does nothing.

The rules are those of all DreamShow's remote controls:

- **GO** in show mode only goes if Go is possible (it is greyed out on the cue window otherwise); in the editor, it acts like the space bar.
- **Previous scene** and **Go to scene no.** play the scene reached in show mode; in the editor, they jump to it without changing the play or pause state.

## Feedback on the LEDs

If the controller has a MIDI output with the same name as its input (the most common case), DreamShow turns on or off the LEDs of buttons learned **on a note**. It sends a note with velocity 127 to turn on, 0 to turn off, on the same channel and the same note number, and only when the state changes (checked four times a second).

| Button action | LED lit when... |
|---|---|

| | |
|---|---|
| **GO** | Go is possible |
| **Go to scene no.** | this scene is the current scene |
| **Blackout** | the blackout is active |
| **LIVE output** | the LIVE button is on |
| **Pause** | the cue list is paused |
| **Play** | the cue list is playing a scene |

The other actions never light an LED, and controls learned on CC receive no feedback. Depending on the controller, the LEDs may not react to these messages; it depends on the device.

## MIDI messages received

At the bottom of the window, **MIDI messages received** lists the latest messages (the 60 most recent, newest at the top), with the time, the controller name and the message ("Note 36 on 100 · ch 1", "CC 7 = 64 · ch 1"…). Handy for checking that a controller is sending something, and what.

## Step by step

### Making pad 1 the GO button

1. Plug in the controller, then open **DMX > MIDI controllers...**.
2. Tick the controller in **Controllers**, then click **Apply**. The status line should show its name.
3. Click **+ Mapping**: a **GO** line appears.
4. Click **Learn**, then hit pad 1. The line shows "Note … · ch …".
5. Test: in show mode, the pad triggers Go; its LED, if it has one, lights up when Go is possible.

### A fader for a fixture's intensity

1. Find the fixture's universe and intensity channel, for example in **DMX > DMX values...** (universe 1, channel 25).
2. Add a mapping, choose **DMX fader (universe.channel)** and type `1.25` in the argument box.
3. **Learn**, then move the controller's fader.
4. Add a second mapping **Give the channels back to the cue list** (empty argument) on a button, to release everything in one gesture.

## Good to know

- To learn a control, the controller must be **ticked and applied**: DreamShow only hears open controllers.

- Mappings are saved in DreamShow's settings (not in the show): they apply to all shows.
- A channel taken by a MIDI fader stays taken until it is released: by a **Give the channels back to the cue list** mapping, by **Release all** or by closing the **DMX faders** window. It goes in front of the general blackout.
- The MIDI commands are the same as those of the OSC remote and the web remote.

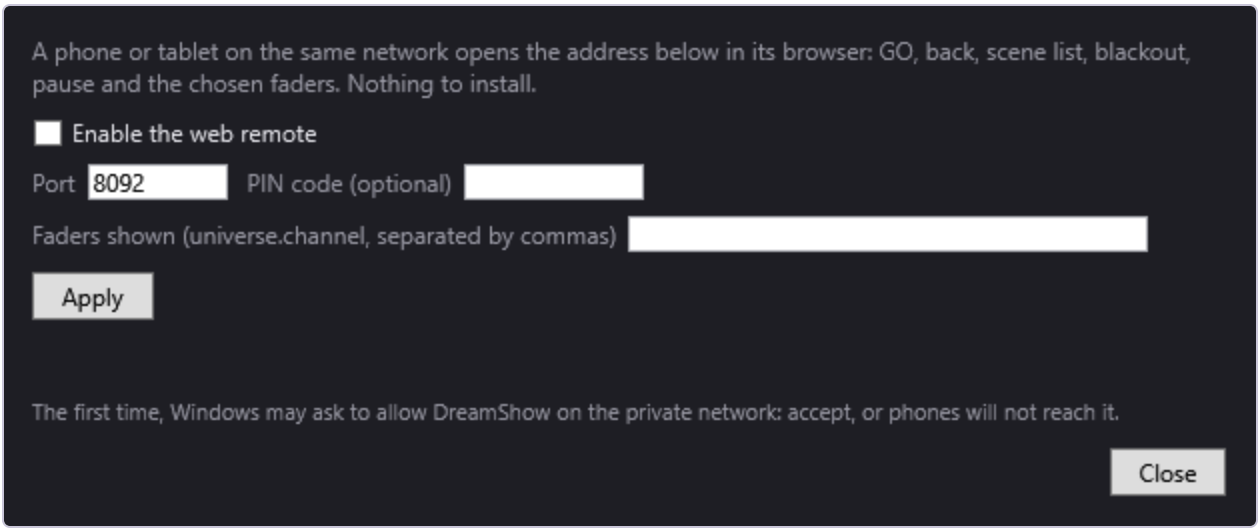
## See also

- Sound-to-light and tempo
- Timecode (LTC / MTC)
- DMX values and DMX faders
- The OSC remote
- Show mode and the cue window

# The web remote

A phone or tablet connected to the same network as the control PC drives the show from its browser: GO, back, scene list, blackout, pause, reset to “Ready”, tap tempo and a few chosen DMX faders. Nothing to install on the phone. Handy for starting the show from the house, the stage or the wings.

**To open it:** menu **DMX > Web remote...**.




## The settings window

- **Enable the web remote**: puts the remote into service. Off by default, because it opens a network port on the PC.
- **Port**: the remote's network port, **8092** by default (from 1 to 65535).
- **PIN code (optional)**: if you type a code, the phone will have to give it before it can read the state or send a command. Empty = no code.
- **Faders shown (universe.channel, separated by commas)**: the DMX channels that will appear as faders on the page, for example `1.6, 1.12` for channels 6 and 12 of universe 1. The separators `,` and `;` are accepted between two channels; up to 16 faders. Empty = no faders.
- **Apply**: saves the settings and (re)starts the remote. **Nothing changes until you have clicked Apply.**
- **The status line**: the address to open on the phone, for example “Web remote: http://192.168.1.10:8092/”; or “Web remote off”; or a message if the port is already used by another program.
- **Close**: closes the window; the remote stays in service.

A note reminds you: “The first time, Windows may ask to allow DreamShow on the private network: accept, or phones will not reach it.”

Once enabled, the remote restarts automatically each time DreamShow is launched.

## The page on the phone

From top to bottom:

- **The current scene**: its number and its name (“Ready” before the first GO).
- **Next: …**: the name of the next scene.
- **GO**: a big button, bright green when GO is possible, dark green otherwise. It follows the same rules as DreamShow's **GO** button: in Show mode, it only goes off if GO is possible; in the editor, it acts like the space bar.
- **◀ Back**: goes back to the previous scene.
- **Blackout**: toggles the general blackout; the button turns red when blackout is active.
- **Pause** / **Play**: pauses the cue list or restarts it; the label says what the button will do.
- **Ready**: goes back to the “Ready” state (blackout, media stopped, the next GO starts scene 1), after the confirmation “Go back to “Ready” (dark stage, media stopped)?”.
- **TAP**: one tempo tap; when a tempo is known, the button shows it (“TAP · 124 BPM”). See Sound-to-light and tempo.
- **Faders** (only if channels are chosen): for each channel, its address and what it drives (fixture · channel), a slider from 0 to 255, the value, and a **×** button to release it. The slider follows the value sent; moving it **takes** the channel, as in **DMX > DMX faders…**, and its label turns orange. **Give the channels back to the cue list** releases all the channels taken.
- **Scenes**: the list of all the scenes; the current scene is highlighted. Touch a scene to go straight to it.

The page asks DreamShow twice a second. If the PC no longer answers (Wi-Fi dropped, DreamShow closed), it displays “DreamShow is not answering”.

The page's labels are in the language of DreamShow's interface.

### The PIN code

When a code is set, the browser asks for it (“Remote PIN code”) as soon as the page is opened, then remembers it. If the code is wrong or has changed, it asks again.

## Step by step

### Starting the show from a phone

1. The phone and the control PC must be on the same network (same Wi-Fi or same router).

2. In DreamShow, open **DMX > Web remote...**, tick **Enable the web remote**, optionally type a PIN code, then **Apply**.
3. Note the address displayed (for example `http://192.168.1.10:8092/`).
4. On the phone, open the browser and type this address. If Windows asked to allow DreamShow on the network, accept.
5. Open Show mode on the PC, then start the show with the phone's **GO** button.

### Adding a fader for the house lights

1. Find the universe and channel of the house lights (**DMX > DMX values...**, by hovering over the cells).
2. In **Faders shown**, type for example `1.120`, then **Apply**.
3. The fader appears on the phone's page after a moment.

## Good to know

- The remote works over plain HTTP, on the local network. The PIN code prevents slips from another phone on the network, but it is not encryption: do not expose it on the Internet.
- If the PC has several network cards, the address displayed is that of one of them; the phone must use the PC's address on its own network.
- Channels taken from the phone override the cue list and the general blackout. They stay taken until **Give the channels back to the cue list**, the **×** button, or the closing of the **DMX faders** window.
- Touching a scene in the **Scenes** list jumps straight to it, without a fade.
- The commands go through the same path as the OSC remote and the MIDI controllers: they behave the same.

## See also

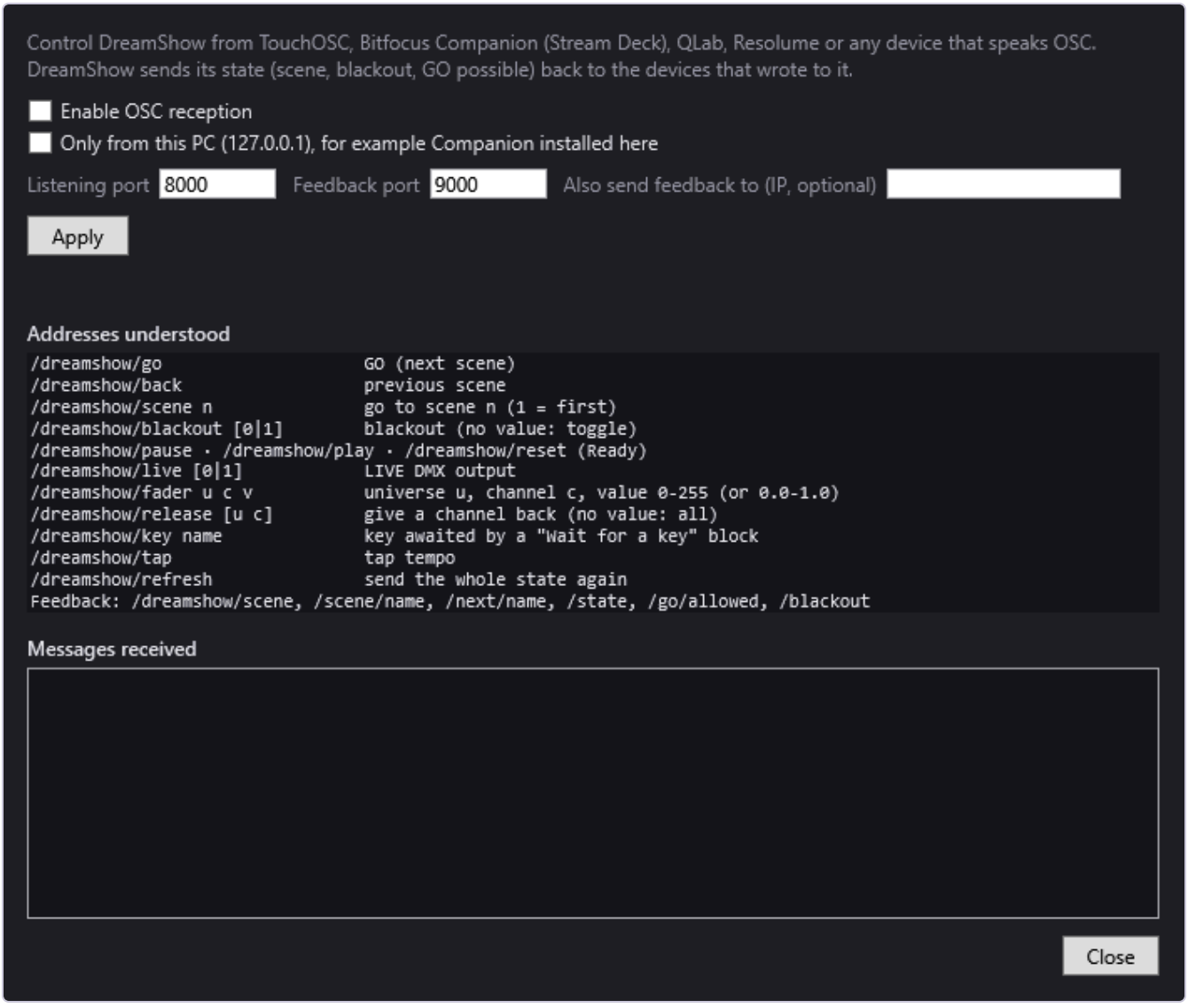
- Show mode and the cue window
- The OSC remote
- MIDI controllers
- DMX values and DMX faders
- The follow spot on a tablet

# The OSC remote

**OSC** (Open Sound Control) is a network command language widely used in show control: tablet apps, programmable button boxes, cue-playback or video software can send OSC messages. DreamShow receives them to fire GO, change scene, go to blackout, force DMX channels… and sends its state (current scene, blackout, GO possible) back to the devices that write to it, so they can display it.

**To open it:** menu **DMX > OSC remote…**.

An OSC message is made up of an **address** that looks like a path ( `/dreamshow/go` ) and, sometimes, **arguments** (numbers or text). It travels over UDP to an IP address and a port.



## The window

- **Enable OSC reception**: puts DreamShow in listening mode. Off by default, because it opens a network port.
- **Only from this PC (127.0.0.1), for example Companion installed here**: only accepts messages coming from a program installed on this same PC. Devices on the network can no longer reach DreamShow, and Windows asks no firewall question.
- **Listening port**: the port on which DreamShow receives, **8000** by default.
- **Feedback port**: the port on which DreamShow sends its state back to the devices, **9000** by default.
- **Also send feedback to (IP, optional)**: an IP address to which the state is sent permanently, even if that device has not sent anything yet. Empty = only the devices that have written.
- **Apply**: checks the values (ports from 1 to 65535, valid IP address), saves them and restarts listening. **Nothing changes before Apply.**
- **The status line**: "OSC off", or "OSC: listening on port 8000 — this PC's address to enter in the device: 192.168.1.10", or "OSC: port 8000 is already in use or refused (…)" if another program occupies that port.
- **Addresses understood**: the reminder of the commands (detailed below).
- **Messages received**: the last messages that arrived (the 60 most recent, the newest at the top), with the time and the IP address of the sender. An unknown `/dreamshow/…` address is flagged there by "unknown address: …".
- **Close**: closes the window; listening carries on.

Once enabled, OSC listening restarts automatically each time DreamShow is launched.

## The commands received

All addresses start with `/dreamshow/`. Upper and lower case do not matter, and a final slash `/` is tolerated. Grouped messages (OSC "bundles") are accepted; their timestamp is ignored, everything is executed at once.

| Address | Arguments | Effect |
|---|---|---|
| `/dreamshow/go` | none, or 1 / 0 | GO (next scene). |
| `/dreamshow/back` | none, or 1 / 0 | Previous scene. |
| `/dreamshow/prev` | none, or 1 / 0 | Same as `/dreamshow/back`. |
| `/dreamshow/scene` | n | Go to scene n (1 = first). |

| | | |
|---|---|---|
| `/dreamshow/blackout` | none, or 0 / 1 | General blackout. With no argument: toggles. With 1: blackout; with 0: end of blackout. |
| `/dreamshow/pause` | none, or 1 / 0 | Pauses the cue list. |
| `/dreamshow/play` | none, or 1 / 0 | Restarts the cue list. |
| `/dreamshow/reset` | none, or 1 / 0 | Reset: “Ready” state. |
| `/dreamshow/live` | none, or 0 / 1 | The editor's **LIVE** button. With no argument: toggles; 1: on; 0: off. |
| `/dreamshow/fader` | u c v | Sets channel c of universe u to value v. |
| `/dreamshow/release` | u c, or none | Gives channel c of universe u back to the cue list; with no argument: gives back all channels. |
| `/dreamshow/key` | name | Releases a “Wait for a key” block that is waiting for this key. |
| `/dreamshow/tap` | none, or 1 / 0 | One tap-tempo tap. |
| `/dreamshow/refresh` | none | Sends the whole state back at once. |

Details:

- **Buttons** ( `go` , `back` , `prev` , `pause` , `play` , `reset` , `tap` ): many devices send 1 on press and 0 on release. DreamShow only acts on the press: with no argument, or with a value of 0.5 or more. A value below 0.5 is ignored.
- **Values accepted**: integers, decimal numbers, numbers written as text, and true / false (true = 1).
- `/dreamshow/scene n` : the number is rounded; a scene that does not exist is ignored.
- `/dreamshow/fader u c v` : universe and channel counted from 1. The value v goes from 0 to 255; if it is sent as a decimal number between 0.0 and 1.0 (as with a tablet fader), it is converted to 0 to 255. The channel is taken as with **DMX > DMX faders...**: it overrides the cue list and the blackout until it is released.
- `/dreamshow/release` with no universe or channel (or with a single number, for example `1` sent by a button) gives back all channels.
- `/dreamshow/key name` : the name is text, the key the block is waiting for, for example `Space` , `A` or `F1` . See Waits.

Behaviour depending on the mode:

- **In Show mode** (cue window open): `go` only goes off if GO is possible (same rule as the greyed-out **GO** button); `back`, `prev` and `scene` jump to the scene without a fade and make it play.
- **In the editor**: `go` acts like the space bar (including the effect of the **Disengage auto-fade** box); `back`, `prev` and `scene` jump to the scene without changing the play or pause state.
- `pause`, `play` and `reset` act in both cases, **even during a performance**.
- `live` changes nothing to the output in Show mode: the output is always active there.

## State feedback

DreamShow sends its state back:

- to **each device that has sent it a message in the last ten minutes**, at that device's IP address and on the **Feedback port** (9000 by default, not the port the message came from);
- and, if it is filled in, to the **Also send feedback to** address.

It sends it after each command received, and as soon as the state changes (checked four times a second). `/dreamshow/refresh` forces a complete send.

| Address sent | Value |
|---|---|
| `/dreamshow/scene` | number of the current scene (integer); 0 = “Ready” |
| `/dreamshow/scene/name` | name of the current scene (text); “Ready” before the first GO |
| `/dreamshow/next/name` | name of the next scene (text); empty on the last scene |
| `/dreamshow/state` | `ready` (Ready), `paused` (paused) or `playing` (playing) |
| `/dreamshow/go/allowed` | 1 if GO is possible, otherwise 0 |
| `/dreamshow/blackout` | 1 if blackout is active, otherwise 0 |

The text “Ready” follows the language of the interface.

## Examples

The steps below describe what DreamShow expects. How to create a button or a message in each application depends on that application: refer to its documentation.

### A tablet with TouchOSC

1. In DreamShow, tick **Enable OSC reception**, leave the port at 8000, **Apply**. Note the PC's address displayed.
2. In the tablet's app, set the OSC connection: host = the PC's address, send port = 8000, receive port = 9000.

3. Create a button that sends `/dreamshow/go`, another `/dreamshow/back`, a third `/dreamshow/blackout`.
4. For an intensity fader, make it send `/dreamshow/fader` with the arguments universe, channel, then the fader's value (0.0 to 1.0).
5. To display the scene name, bind a label to the address `/dreamshow/scene/name`; to light the GO button when GO is possible, bind it to `/dreamshow/go/allowed`.

### A button box with Companion on the same PC

1. In DreamShow, tick **Enable OSC reception** and **Only from this PC (127.0.0.1), for example Companion installed here**, then **Apply**.
2. In Companion, set up OSC sending to host 127.0.0.1, port 8000.
3. Assign each key a message: `/dreamshow/go`, `/dreamshow/back`, `/dreamshow/blackout`, `/dreamshow/scene` with the scene number…
4. For Companion to receive the state, make it listen on the feedback port (9000); DreamShow replies to it as soon as it has sent a first message.

### Firing GO from a QLab cue list

1. In DreamShow, enable OSC reception on port 8000 (without ticking “Only from this PC” if QLab runs on another computer).
2. In QLab, create a network cue that sends the OSC message `/dreamshow/go` to the DreamShow PC's address, port 8000.
3. Place this cue in QLab's cue list: each time it goes off, DreamShow does a GO (if GO is possible in Show mode).
4. To go to a specific scene: `/dreamshow/scene 5`.

   The names of menus and settings change from one version of TouchOSC, Companion or QLab to another: refer to their documentation to find where to enter the address, the port and the OSC message. On the DreamShow side, only the PC's address, the port (8000 by default) and the messages listed above matter.

## Good to know

- On first use, the Windows firewall may ask you to allow DreamShow on the network: accept, unless you only use **Only from this PC**.
- The **Messages received** list lets you check that a device is sending, and what: if nothing appears, the problem is on the network side (IP address, port, firewall).
- The same commands are used by MIDI controllers and the web remote.
- Sound-to-light also sends its automatic GOs by this same path: they obey the same rules.

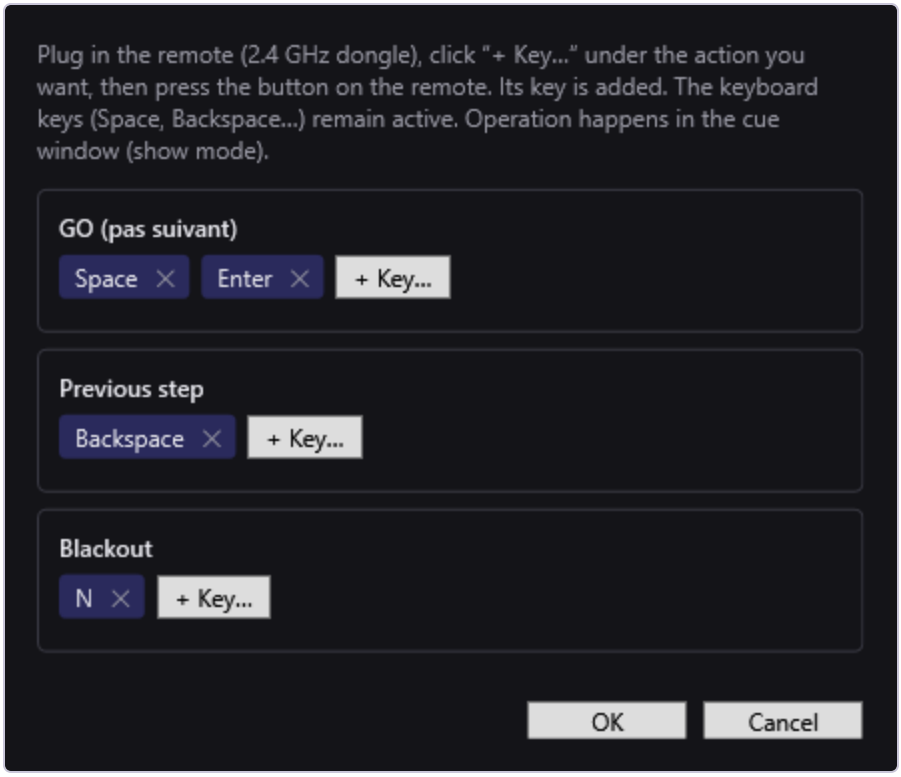
## See also

- The web remote
- MIDI controllers
- Show mode and the cue window
- DMX values and DMX faders
- Waits (key, time, date, sun)

# The remote (GO keys)

This window chooses the keys that drive the cue window (Show mode): **GO**, **Previous step** and **Blackout**. It is mainly used to connect a **wireless presentation remote** (the small box with a 2.4 GHz USB dongle used to step through slides): such a box behaves like a keyboard and sends keys (Page Down, arrows…). You just have to teach DreamShow which key each button sends.

**To open it:** menu **File > Remote (GO keys)…** (tooltip: “Assign the keys of a 2.4 GHz remote to the GO / Previous step / Blackout actions of the cue window.”).




## The window

The text at the top summarises what to do: plug in the remote, click “+ Key…” under the action you want, then press the remote's button; its key is added. The keyboard keys stay valid. Operation happens in the cue window (Show mode).

Three frames, one per action:

- **GO (pas suivant)** (GO, next step): starts the next scene (GO).
- **Previous step**: goes back to the previous scene.
- **Blackout**: toggles the general blackout.

In each frame:

- **The keys already assigned**, as pills. Each pill carries the key's name and a ✕ (Remove) to take it off.
- **+ Key…**: click, the button becomes “Press a key…”; then press the keyboard key or the remote's button. The key is added to the list. **Esc** cancels the capture without adding anything. A key already in the list is not added twice.

An action can have several keys: they all work.

At the bottom:

- **OK**: saves the keys. If the **GO (pas suivant)** list is empty, the **Space** key is put back in it automatically: GO is never left without a key.
- **Cancel**: closes without changing anything.

### The default keys

| Action | Default keys |
|---|---|
| GO (next step) | Space, Enter |
| Previous step | Backspace (the back-erase key) |
| Blackout | N |

### The names displayed

Some keys have a plain name: **Space**, **Enter**, **Esc**, **Backspace**, **Page ↓**, **Page ↑**, **← Left**, **Right →**. The others appear under their technical name (for example `B`, `F5`, `Down`). Presentation remotes often send **Page ↓** for “next” and **Page ↑** for “previous”.

## In the cue window

- The big buttons of the cue window remind you of the **first** key of each action: “GO (Space)”, “◀ Previous (Backspace)”, “BLACKOUT (N)”. These labels update the next time Show mode is opened; the keys themselves work at once.
- The keys only work when **the cue window has focus** (it is the active Windows window). Click once on its title bar if another window has taken over.
- **GO** only acts if GO is possible (see Show mode and the cue window).
- If a **Wait for a key** block is waiting, the key it expects is first used to release it, before any other action. See Waits.

## Step by step

### Connecting a presentation remote

1. Plug in the remote's USB dongle and switch it on.
2. Open **File > Remote (GO keys)...**.
3. Under **GO (pas suivant)**, click **+ Key...**, then press the remote's “next” button. A pill appears (often **Page** ↓).
4. Under **Previous step**, click **+ Key...** and press the “previous” button (often **Page** ↑).
5. If the remote has a “black screen” button, assign it in the same way to **Blackout**.
6. Click **OK**.
7. Open **Show mode**, click on the cue window, and test each button.

### Preventing a keyboard slip

If the Enter or Backspace key risks being pressed by mistake during the show, remove it with its ✕, then **OK**.

## Good to know

- These keys only apply in the cue window. In the editor, only the **Space** bar fires GO, whatever the list.
- In the cue window, **F9** (compact) and **F11** (full screen) are read before the keys in the list: a remote that sends F9 or F11 cannot be used for GO.
- If the same key is assigned to two actions, GO wins, then Previous step, then Blackout.
- The keys are saved in DreamShow's settings, not in the show: they apply to all shows.
- To drive the show from a phone, a MIDI controller or an OSC app, see The web remote, MIDI controllers and The OSC remote.

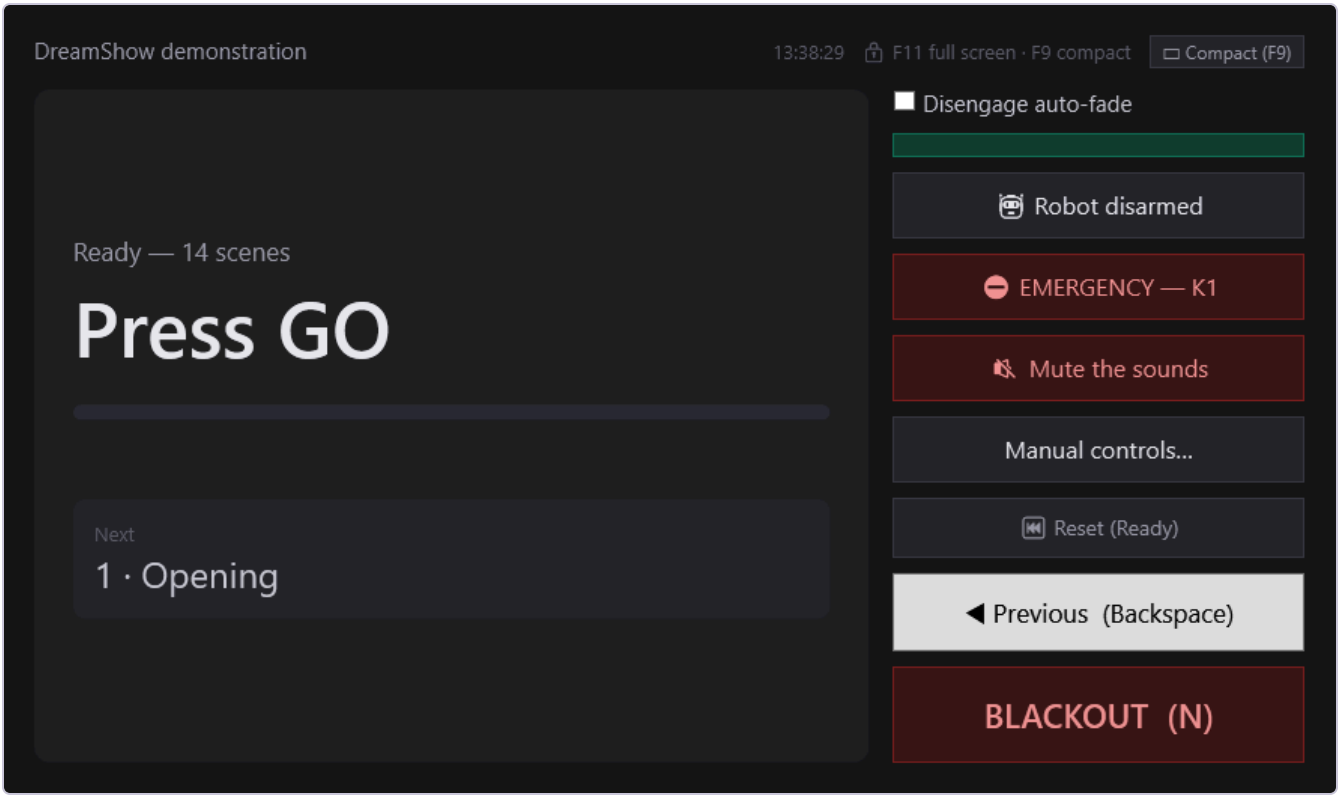
## See also

- Show mode and the cue window
- Waits (key, time, date, sun)
- Keyboard shortcuts
- The web remote

# Show mode and the cue window

**Show mode** is the way you play the show. It opens the **cue window**: a clean screen, readable from a distance, that shows the current scene, its progress, the next scene and the message for the stage manager, with big **GO**, **◀ Previous** and **BLACKOUT** buttons. This window never modifies the show: it only sends orders to the playback engine. It can go onto a second screen, or shrink into a small strip (**Compact**) to keep the editor visible.

**To open it: Show mode** button at the top right of the main screen (tooltip: “Opens the cue window (a second screen is possible)”).




## Opening show mode

On clicking **Show mode**:

- **Clean start** (the usual case): the cue list goes back to the **“Ready”** state: blackout, media stopped, no scene in progress. The first Go will start scene 1 from the beginning. Careful: if a scene was lit in the editor, the venue goes to blackout as soon as it opens.
- **Resume**: if the cue window had been closed in the middle of a show, reopening it resumes **exactly where the show was**: same scene, same position in its timeline, playing.

The window opens in front of the editor and stays there (a click on the editor does not hide it). If you left it compact last time, it opens compact.

During show mode, **the DMX output is always active**, even if the editor's **LIVE** button is off: a stray click on LIVE cannot plunge the show into darkness. Only visualiser mode cuts the output, even in show mode.

## The cue window

### The top banner

- On the left, **the show's name**.
- On the right, an information line:
  - before the first Go: the time;
  - after: "21:04:12 · start 20:31:05 · length 0:33:07", that is, the time, the time of the first Go and the length elapsed since;
  - at the end of the show, the length freezes and "· END 22:15:40" is added (see below);
  - then the reminder " 🔒 F11 full screen · F9 compact";
  - when parallel sequences are running: "⇉ in parallel: …" with their names (see Parallel sequences).
- ▭ **Compact (F9)**: shrinks the cue window into a small strip (see below).

### The current scene

- **The counter**: "Ready — 12 scenes" before the first Go, then "In progress — 3/12".
- **The scene's name**, in very large type. Before the first Go: "Press GO".
- **The progress bar**:
  - during an incoming fade, it shows the fade's progress, and the time shows "fade 2.4 s" (the remaining time);
  - afterwards, it shows progress through the scene's timeline, and the time shows "1:12 / 3:40" (position / length of the timeline);
  - if the scene has nothing to play (empty timeline, or only instantaneous items), the bar stays empty and the time only shows how long you have been on the scene.
- After the time may be added:
  - "↻ loop 2" for a scene that loops (the pass number), with "(max 300 s)" if the loop has a maximum length;
  - "→ fade 3 s": the fade that will be used to **leave** this scene for the next.
- **The cue message**, in yellow: the text entered in the **Cue message:** field of the scene settings ("what to expect before pressing GO"). See Scene settings. When a **Wait for a key** block is waiting, this message is replaced by the block's instruction: "⌨ Press "Space"", or the block's message followed by "⌨ "Space"". See Waits.

- **Next**: the number and name of the next scene, with “(auto)” if it starts by itself after a delay, or “(chained)” if it chains at the end of the current scene. On the last scene: “— END OF SHOW — last scene reached at 22:15:40”, or “— end of show —”.

### The buttons

- **Disengage auto-fade** (tick box): “Ticked: no scene chains on its own (auto-fade / chained) — you advance everything with GO.” Automatic triggers (after a delay, chained) are ignored; each scene change is made with Go. The box is the same as the editor timeline's: ticking one ticks the other.
- **GO (Space)**: starts the next scene. The first Go from “Ready” starts scene 1 and marks the **start of the show**. The button is **greyed out** (and Go has no effect, by click or by keyboard):
  - on the last scene;
  - as long as the current scene's timeline has not finished (incoming fade in progress, or the red line has not yet reached the end of the last item). Exception: a scene **on loop** accepts Go at any time (this is how you get out of a loop). Go starts with the fade of the scene being left; the new scene's timeline starts once this fade has finished. See Fades and mixes.
- 🤖 **Robot disarmed** / 🤖 **Robot ARMED**: visible only if the venue declares robots. Arms or disarms the robots: when disarmed, the timeline's Robot blocks are ignored. The button turns red once armed, and shows the first robot's state (mode, battery, “⚠ AU SOL” (ON THE FLOOR), “hors ligne” (offline)…).
- ⛔ **EMERGENCY — robot name**: one red button per declared robot, and ⛔ **EMERGENCY — ALL** first if there are several. “Emergency stop: immediate stop then motors off (the robot slumps). Goes through even when disarmed.” See Humanoid robots.
- 🔕 **Mute the sounds**: “Immediately stops all playing sounds.” Same action as the button of the editor's **Audio** panel. This button only stops the sounds played by DreamShow: a video's sound carries on as long as the video plays.
- **Manual controls…**: opens the manual controls (winch, power supply, shutters), identical to the editor's. See Manual controls.
- ⏮ **Reset (Ready)**: “Back to the pre-show state: blackout, media stopped. The next GO starts scene 1.” Parallel sequences stop, waits in progress are cancelled, and the start and end of the show are cleared. Fixtures set to **hold** (for example a winch) keep their value. There is no confirmation.
- ◀ **Previous (Backspace)**: goes back **instantly** (without a fade) to the start of the previous scene, and plays it. No effect on scene 1 or before the first Go.
- **BLACKOUT (N)**: toggles the general blackout. The button turns bright red as long as the blackout is active; another press lights up again. During the blackout, the cue list keeps running behind the scenes (timeline, sounds), only the fixtures are at zero, with the exception of:
  - fixtures set to **hold** in the venue (field “If not mentioned in a scene”), so that a blackout never moves a winch;

    - the manual controls, the follow spot on a tablet and the channels taken by the DMX faders, which go in front of the blackout.

The keys in brackets are those set in **File > Remote (GO keys)…** (see The remote (GO keys)).

### Keyboard keys

When the cue window has focus:

| Key | Effect |
|---|---|
| Space, Enter (by default) | GO |
| Backspace (by default) | ◀ Previous |
| N (by default) | BLACKOUT |
| F9 | Compact / full window |
| F11 | Full screen / normal window |

If a **Wait for a key** block is waiting, its key is used first to release it.

## The compact strip (F9)

▭ **Compact (F9)** (“Shrinks the conductor to a small always-on-top strip: the rest of the screen is visible again. F9 to come back.”) folds the cue window into a narrow strip, **always in front**. Useful on a single screen, to keep the timeline or the simulation visible. The strip shows:

- “3/12 scene name”, and the time on the right;
- a thin progress bar;
- the cue message (in yellow) if there is one;
- “ ▸ ” followed by the next scene;
- three buttons: ◀ (Previous scene), **GO** and **BLACKOUT**, which do exactly the same as in the full window;
- ⤢ (Full window (F9)) to return to the full window.

The first time, the strip is placed at the top right of the screen. Afterwards, each presentation (full or compact) returns to where you left it, as long as the cue window stays open. The full / compact choice is remembered for the next opening of show mode. Going to full screen (F11) leaves the compact strip.

## On a second screen

The cue window is a separate window: drag it by its title bar onto the second screen (the stage manager's), then press **F11** so that it fills that whole screen. **F11** again puts it back in a normal

window. The editor stays on the main screen.

## What stays active in the editor during the show

The editor is **not** locked during show mode. What you need to know:

- **The scene list follows the show**: the current scene is selected automatically at each change, and the timeline's red line advances with it.
- **Clicking another scene in the list, with the LIVE button on, makes the show jump to that scene** (and play it). Avoid clicking in the list during a performance.
- **A change to the current scene applies at once on stage** (colours, intensities, blocks…), without restarting the red line.
- The editor's **space bar** does a Go, with the same rules as the cue window's **GO** button.
- Still usable: the **Mute the sounds** button of the **Audio** panel, the robot arming button, **Venue > Manual controls…**, the **DMX** menu (values, faders, sound-to-light, timecode…), the web, OSC and MIDI remotes.
- The **LIVE** button has no effect on the output during show mode (it stays active).
- The editor's **Disengage auto-fade** box and the cue window's are the same.

## The length and the end of the show

- **Start**: the first Go given from the “Ready” state. The time is shown in the banner (“start 20:31:05”), and the length runs from then on.
- **End of the show**: arriving on the **last scene**, by a Go or by automatic chaining. At that moment:
  - the length freezes and the banner adds “· END 22:15:40”;
  - the **Next** frame shows “— END OF SHOW — last scene reached at 22:15:40”;
  - the log notes “END OF SHOW: last scene reached — length 1:44:35”;
  - the **GO** button is greyed out. The last scene plays its timeline normally to the end.
- Arriving on the last scene by a jump (◀ Previous, a click on a scene, a jump programmed in the timeline, a “go to scene” remote command, timecode) does not mark the end: **Next** then simply shows “— end of show —”.
- ⏮ **Reset (Ready)** clears the start and end: the length will start again at the next first Go.

## Closing the cue window

Close the window (cross in the title bar). **There is no confirmation.**

- DreamShow remembers where the show had got to: reopening **Show mode** resumes it exactly at that scene and that position.
- The editor takes over again: the selected scene is displayed, transport stopped (its final state stays visible).

- If the editor's **LIVE** button is off, **the DMX output stops** as soon as the cue window is closed.

To start from zero on reopening, click ⏮ **Reset (Ready)** before closing the cue window.

Quitting DreamShow during show mode asks for confirmation: “A show is RUNNING. Quit DreamShow anyway?”. In case of a crash or a power cut, see Recovering after a crash or power cut.

## Step by step

### Playing a performance

1. Before the audience comes in, open the show and check that **LIVE** is on.
2. Click **Show mode**. The cue window shows “Ready — N scenes” and the venue is in blackout.
3. Drag the cue window onto the second screen and press **F11**.
4. On the cue, press **GO** (or Space, or the remote's button): scene 1 starts and the length begins.
5. Follow the cue message (in yellow) and the next scene; when **GO** becomes bright again, the next Go is possible.
6. In case of trouble: **BLACKOUT**, **Mute the sounds**, or ◀ **Previous** to replay the previous scene.
7. On the last scene, the cue window shows “— END OF SHOW —” and the total length. Close the cue window.

### Preparing the next performance

1. Reopen **Show mode**: the cue window resumes where it was.
2. Click ⏮ **Reset (Ready)**: blackout, media stopped, the cue window shows “Ready”.
3. The next **GO** will restart scene 1 and a new length.

### Keeping the editor visible on a single screen

1. In the cue window, press **F9**: it becomes a strip at the top right.
2. Place the strip where it does not get in the way; it stays in front.
3. The ◀, **GO** and **BLACKOUT** buttons and the keys stay active. **F9** or ⤢ to return to the full window.

## Good to know

- A slow PC can be relieved by switching off the simulation rendering (⏻ button of the simulation panel): the DMX output is not affected. See 2D and 3D simulation.
- The cue window never changes the show; but the editor, for its part, stays editable during the show (see above).
- The cue window's keys only work if it has focus. After a click in the editor, click on the cue window again.

- The remotes (web, OSC, MIDI) follow the same Go rules as the cue window. Careful: their **Pause** and **Ready** commands also act during the show.

## See also

- The remote (GO keys)
- Recovering after a crash or power cut
- Scene settings
- The scene timeline
- Waits (key, time, date, sun)
- Parallel sequences
- Manual controls
- Humanoid robots
- Fades and mixes

# Recovering after a crash or a power cut

If DreamShow stops abruptly (crash, forced shutdown, power cut), it offers on restart to pick up exactly where it stopped: same show, same scene, same second of the timeline. In the middle of a performance, you relaunch DreamShow, answer **Yes, resume**, and the cue list starts again.

**To open it:** nothing to do. The offer appears by itself at the start-up that follows an abnormal stop.

## How it works

While DreamShow is running, about once a second it notes a small **resume marker**: the open show, the scene, the position in its timeline, and whether it is a performance (cue window open) or editing work.

- **During a performance** (cue window open, see Show mode and the cue window): it is the scene actually being played and its position that are noted.
- **When editing**: it is the scene selected in the **Scenes** panel that is noted, with its position if playback is on that scene (otherwise the start of the scene).

The marker is erased when DreamShow closes normally (after your **Yes** answer to "Quit DreamShow?"). It is also erased when you close the cue window or open another show. Only an abnormal stop leaves it in place.

Writing the marker never slows the show down: if it fails, DreamShow carries on without saying anything.

## On restart

Once the main window is displayed, if a marker less than 6 hours old is found, DreamShow opens the **Resume after an interruption** window, always on top (it comes in front of the other open programs). It shows:

- "DreamShow stopped during a show" (or the equivalent message for an editing session), with the time of the last point noted;
- **Show:** the file name of the show;
- **Scene:** the scene number;
- **Position:** the position in the timeline (for example "1:12.50" for 1 minute 12.5 seconds, or "8.20 s");
- the question "Resume exactly at that point?".

The buttons:

- **Yes, resume** (Enter key):
    1. DreamShow opens the show concerned if it is not already open;
    2. selects the scene;
    3. opens the cue window;
    4. places playback on the scene and second noted and **plays** from there. The light and the sound are recalculated for that moment: a sound in progress resumes at the right place in its file.
- **No** (Esc key): the marker is erased and DreamShow starts normally.

The resume comes before the **Welcome to DreamShow** start screen.

If the scene noted no longer exists in the show (the show has been modified and saved since), the resume is abandoned and the **Log** says so: "Cannot resume: the scene no longer exists in this show."

A marker more than 6 hours old is ignored and erased: a crash from the day before does not ask you a question the next day. An unreadable marker is also ignored. These cases are noted in the **Log**.

## What is resumed, what is not

Resumed:

- the show, the scene and the position in the timeline;
- everything the timeline produces at that moment: light, effects, sound, video;
- a wait in progress (a time, a date, sunrise or sunset, a key): it is replayed, and if its time went by during the interruption, the show carries straight on (the **Log** says so).

Not resumed:

- changes to the show that had not been saved: the show is reopened from its file. Save often (Ctrl+S);
- adjustments made by hand live during the performance, which are not part of the marker:
    - channels forced in **DMX faders**;
    - the general **BLACKOUT**: the light comes back on restart;
    - parallel sequences in progress;
    - moving heads driven from the follow-spot tablets (the follow-spot server is restarted from the menu);
- a sound started in a previous scene that was still playing: only the sounds of the resumed scene are restarted;
- the pan and tilt of moving heads set by a previous scene, if the resumed scene does not set them: these moving heads go back to their rest position;
- a sound level set by a previous scene.

The **manual controls**, on the other hand, are kept: they are saved in the venue at each gesture and reapplied at start-up.

## Step by step

### Resuming a show after a power cut

1. Switch the PC back on and relaunch DreamShow.
2. The **Resume after an interruption** window appears: check the show, the scene and the position.
3. Click **Yes, resume** (or press Enter).
4. The cue window opens and the show restarts at the position noted. Carry on with the cue list as normal with **GO**.

### Starting from scratch anyway

1. In the **Resume after an interruption** window, click **No**.
2. DreamShow starts normally, on the last show opened.

## Good to know

- A **Yes** always restarts playback in the cue window, even if DreamShow had stopped during editing. If you were preparing the show and do not want it to start playing, answer **No**.
- An error in the interface does not stop DreamShow: it is noted in the **Log** and the software carries on, so that the stage keeps its light.
- The marker is stored in the file `recovery.json` in the `DreamShow` folder of your Windows account's local data (`%LOCALAPPDATA%\DreamShow`). It is shared by all the copies of DreamShow launched by this account on this PC. A copy launched with the `--no-recovery` option neither reads nor writes this marker (useful for a test copy next to the control PC).
- If crashes keep happening, note the version (**? > About DreamShow...**) and copy the **Log** (the panel's **Copy** button) for support. See Troubleshooting.

## See also

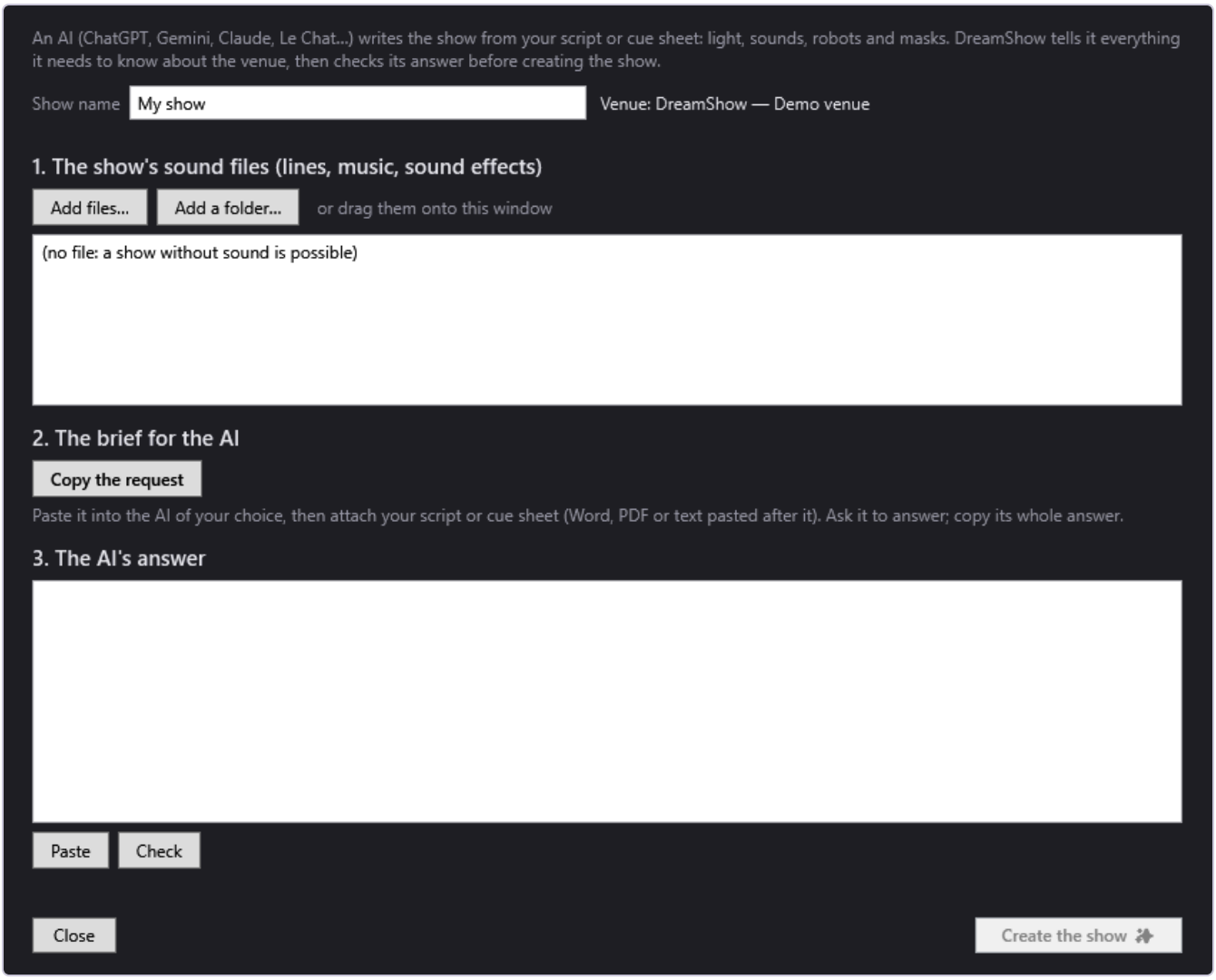
- Show mode and the cue window
- Shows, files and projects
- Log, Audio and Clock
- Troubleshooting

# Creating a show with an AI

You already have the script of a play, a cue sheet or a running order: a conversational AI (ChatGPT, Gemini, Claude, Le Chat…) writes the show from that document, with the light, the sounds, and, if the venue has any, the robots and LED masks. DreamShow gives it everything it needs to know about your venue, then checks its answer line by line before creating the show. The AI never writes DMX or addresses: it chooses moods, zones, groups and files, and DreamShow builds the rest with your real fixtures.

**To open it:** menu **File > Create a show with an AI…**, or, in the first-show wizard, the choice **With an AI, from my script**.

The window stays open while you work in your browser: you can switch between the two as many times as necessary. A venue must be loaded; otherwise DreamShow displays “No venue loaded.”.



## The window header

- **Show name**: the name of the show to create. Suggested value: “My show”. This name is also used to name the show's file and its media folder (see below). As soon as the media folder is created (first file added, first request copied or first check), the box becomes read-only; its tooltip then says “Set by the show's media folder: …” followed by the folder's path.
- **Venue: …**: reminds you of the active venue. This is the one DreamShow describes to the AI and the one the show will use. To work with another venue, change venue (menu **Venue > Change venue**) before opening this window.

## 1. The show's sound files (lines, music, sound effects)

Here you give DreamShow all the sounds the AI may place in the show. A show without sound is possible: the list then displays “(no file: a show without sound is possible)”.

- **Add files…**: opens a file chooser (several at a time). Formats offered: wav, mp3, m4a, aac, wma, flac.
- **Add a folder…**: adds all the sound files of a folder.
- **or drag them onto this window**: you can also drag files from Windows Explorer onto the window.

Files that are not sounds are ignored. The sounds kept are **copied** into the show's media folder (`medias\<show name>` in DreamShow's data folder); a file of the same name already there is replaced. The log notes “… sound file(s) added to …”.

The list shows each file preceded by ♪, with its length in seconds when DreamShow can read it, for example `♪ acte1-scene2.mp3 (12.4 s)`. These lengths are passed to the AI: it uses them to chain the lines.

> The WAV files that DreamShow makes itself for a mask's mouth (see below) do not appear in the list: the AI always designates the original file.

## 2. The request for the AI

- **Copy the request**: places in the clipboard a complete text intended for the AI. The message “Copied! Paste it into your AI, with your script.” is displayed next to the button.

Then, as the window says: “Paste it into the AI of your choice, then attach your script or cue sheet (Word, PDF or text pasted after it). Ask it to answer; copy its whole answer.”

### What the request contains

The request is written in DreamShow's language (French if the interface is in French, English otherwise). It includes:

- **The role**: the AI is a lighting and sound operator writing the show's technical cue list, and must answer **only** with a JSON object, with no text around it.
- **The venue**: its name, the width and depth of the stage, and the reminder that “jardin” is the left side seen from the audience and “cour” the right side.
- **The fixture groups** of the venue, with their exact name, the number and model of the fixtures, their role and their location:
  - role: “moving heads (movement, colour)”, “colour fixtures”, “white, intensity only”, “strobes”, or “SPECIAL DEVICE (smoke, fan, UV, machine…)”;
  - location: “on the floor”, “front of house”, “upstage / backlight”, “over the stage” or “position unknown”. Empty groups are not listed.
- **The robots** of the venue (if it declares any) with all their actions, their usual length, the dances, kicks and gestures available, and the reminder that the robot must be prepared before moving.
- **The LED masks** of the venue (if it declares any) with their actions and their faces, and how to make a character speak.
- **The sound files available**, with their exact name and length.
- **The lighting vocabulary**: moods, zones, group settings, effects (detailed below).
- The sentence “NEVER use the strobes unless the script explicitly asks for it.”
- **The answer format**, with a template to fill in.
- A last section “My script / cue sheet:” where you attach or paste your document.

## 3. The AI's answer

- The large text area receives the AI's answer. You can paste it by hand (Ctrl+V) or with the button below.
- **Paste**: replaces the content of the area with the clipboard text, then starts the check at once.
- **Check**: analyses the answer and displays the report under the buttons.

DreamShow tolerates the text the AI sometimes adds around its answer (introductory sentences, ``` `json tags): it takes everything between the first brace { and the last }` ```.

Any change to the answer disables the **Create the show** ✨ button: you have to check again.

### The check report

The report lists, in order:

- the **errors**, in red, preceded by ✕: they prevent the show from being created;
- the **warnings**, preceded by •: DreamShow has corrected or ignored something, and tells you;
- a summary: “… scenes, … sounds, … robot actions, … mask actions, … effects.”;

- then either “Everything is understood: you can create the show.”, or “Fix the errors (or ask the AI to fix them by pasting these lines back to it), then check again.”

Each line starts with the scene concerned, in the form “Scene 3 “The living room”: …”.

**Errors (creation impossible):**

| Message | Meaning |
|---|---|
| No JSON object in the answer. | The pasted text contains no block between braces. |
| Unreadable JSON: … | The answer is malformed (often cut off: ask the AI to send it again in full). |
| No scene in the answer. | The list of scenes is empty. |
| …: missing file. | A sound, a robot line or a mask speech names no file. |
| …: file “…” is not in the show's media. | The file was not added in part 1. |
| …: robot “…” is not in the venue. | The name matches no robot declared in **Venue > Robots…**. |
| …: unknown robot action “…”. | The action does not exist. |
| …: unknown dance “…”. / unknown move “…”. / unknown gesture “…”. | The choice does not exist for this robot. |
| …: mask “…” is not in the venue. | The name matches no mask declared in **Venue > LED masks…**. |
| …: unknown mask action “…”. | The action does not exist. |
| …: unknown face “…”. | The face is not in the catalogue. |

**Warnings (the show can be created):**

| Message | What DreamShow does |
|---|---|
| Format “…” instead of “dreamshow-show-plan”: read anyway. | It reads the answer regardless. |
| …: “file” understood as “other file”. | The name given by the AI only matches apart from the extension (for example `scene1.wav` for `scene1.mp3`): the real file is used. |
| …: unknown mood “…”, “day” used. | The Day mood is applied. |

| | |
|---|---|
| …: unknown zone "…", "all" used. | The whole stage is lit. |
| …: unknown trigger "…", Go used. | The scene will wait for the Go. |
| …: unknown group "…", ignored. / …: unknown group "…", effect ignored. | The setting or the effect is not created. |
| …: unreadable colour "…". | The colour is not applied (the expected format is `#RRGGBB`). |
| …: the special device "…" is switched on (smoke, fan, UV…): check that this is intended. | The setting is applied, but check it. |
| …: no effect on the special device "…", ignored. | No effect is placed on a special device. |
| …: "circle" needs moving heads, the group "…" has none. | The circle and eight effects only apply to moving heads. |
| …: unknown effect "…", ignored. | The effect is not created. |
| Strobes ("…") are never fired from an AI plan: set them by hand if the show calls for it. | Any lighting or effect requested on a strobe group is removed. |

## Creating the show

- **Create the show** ✨: only active after a check with no errors. DreamShow saves the show in the data folder under the name `<show name>.show.json` (if that name is already taken, it adds "(2)", "(3)"…), associates it with the active venue, then opens it. If the show open until then has unsaved changes, DreamShow first offers to save it. The show created joins **File > Recent shows**, the simulation switches to 3D and the window closes. The log notes "Show "…" created from the AI's answer: … scenes".
- **Close**: closes the window without creating anything. Files already copied stay in the media folder.

## What is created

For each scene of the answer, DreamShow creates a scene of the show, in the same order.

**The final state (the light).** DreamShow builds the light as for a theatre cue list:

- each colour fixture receives the colour and level of the requested mood; the fixtures at the back (backlights, floor devices) have their own tint;

- white fixtures with intensity only cannot change colour: for Night, Mystery and Evening, they simply drop in intensity;
- with a zone other than “whole stage”, the fixtures whose beam falls outside the zone drop to 12 % of their level;
- the moving heads really aim at the zone (with their calibration if they have one, otherwise by calculation from their position), and for “whole stage” they spread across the width;
- in a blackout or house-open scene, the moving heads stay off but already turn towards the zone of the next lit scene (they preset in the dark);
- fixtures marked out of service in the venue are left aside;
- special devices (role “other”: smoke, fans, UV, relays, devices marked “no direction”, “invisible” or manual) are never switched on by a mood;
- then, if the AI gave group settings, they apply on top of the mood; with “only”, all the fixtures outside the groups named are set to 0.

**The scene settings.**

- **Trigger:** “go” gives **Manual GO**, “auto:N” gives **Auto** with a delay of N seconds, “chain” gives **Chained**.
- **Fade to the next (s):** in DreamShow, the fade belongs to the scene being left. The “fade” the AI gives a scene (fade in) is therefore placed on the scene that precedes it. The “fade” of the first scene is not used; the last scene has a fade of 0 s. Values are limited to 0–600 s.
- **Cue message:** receives the AI's “note”, that is what the operator waits for before pressing Go for the next scene.

**The timeline tracks**, created only if they have content:

- **Sound**: one block per file, at its start, with its real length (3 s if the length is unknown);
- **Robot**: one “Robot…” block per action, with the action's default length, or the file's length for “Say” (at least 0.5 s);
- **Mask**: one “Mask…” block per action (2 s by default, or the file's length for a speech);
- **Effects**: one dynamic-effect block per effect, with a fade in and fade out of 0.5 s. With no length given, the effect lasts until the end of the last sound, robot or mask of the scene (30 s if the scene has none).

The six possible effects become these dynamic effects:

| Effect requested | Dynamic effect created |
|---|---|
| rainbow | rainbow on the group, 6 s cycle |
| chase | intensity chase, 1.2 s cycle |
| pulse | intensity pulse (square, 15 % floor), 1 s cycle |
| wave | intensity wave (sine, 30 % floor), 3 s cycle |

| | |
|---|---|
| circle | circle of the moving heads, 4 s cycle (moving heads only) |
| eight | figure of eight of the moving heads, 4 s cycle (moving heads only) |

These cycle lengths correspond to speed 1; the requested speed (“speed”) divides them, between 0.2 and 5. A colour is only taken into account for chase, pulse and wave. Each block created can then be modified by double-clicking, like any dynamic effect (see Dynamic effects).

**The masks' mouths.** The lip-sync of an LED mask is calculated from a WAV file. When the AI makes a mask speak on an m4a, mp3 or other file, DreamShow makes, as soon as the check is done, a WAV of the same name in the media folder (it makes it again if the original file is more recent), and the Mask block uses that WAV. The Sound block keeps the original file. See LED masks.

## The vocabulary the AI uses

You do not have to write this format yourself, but knowing it helps you to read an answer or to ask for a precise correction.

- **Moods** ( `ambience` ): `full` (full light), `day` , `evening` , `night` , `warm` (warm interior), `cold` (cold exterior), `mystery` (tight moving heads), `audience` (house light), `black` , `party` (party: full light, to be completed with effects). With no lighting setting, a scene is in the Day mood over the whole stage.
- **Zones** ( `zone` ): `all` (whole stage), `jardin` (stage right), `centre` (or `center` ), `cour` (stage left), `front` (downstage), `back` (upstage).
- **only**: list of the only groups used by the mood.
- **groups**: exact setting of a group: `group` , `dimmer` from 0 to 1, `color` in the format `#RRGGBB` .
- **effects**: `type` , `group` , `start` , `duration` , `speed` (1 = normal, 2 = fast), `color` .
- **sounds**: `file` and `start` (in seconds from the start of the scene).
- **robot**: `robot` (name), `action` , `param` or `value` depending on the action, `file` for “say”, `start` . See Humanoid robots.
- **mask**: `mask` (name), `action` , `param` , `value` or `file` , `start` . To make a character speak, the AI puts the file in `sounds` (it goes through the PA) and a `say` mask action with the same file and the same start.
- **trigger**: `go` , `auto:N` , `chain` . **fade**: fade in, in seconds. **note**: message for the operator.

Example of a minimal answer:

```
{
  "format": "dreamshow-show-plan", "version": 1, "name": "Le Petit Prince",
  "scenes": [
    { "name": "Accueil", "trigger": "go", "fade": 3, "note": "Go quand la salle est pleine",
      "light": { "ambience": "audience" } },
    { "name": "Le désert", "trigger": "go", "fade": 4, "note": "Go sur « Dessine-moi un mout
      "light": { "ambience": "day", "zone": "centre" },
```

```
      "sounds": [ { "file": "vent.mp3", "start": 0 } ] }
  ]
}
```

## Safety rules

- The AI only describes intentions (mood, zone, group, file, action): it never writes a DMX value or an address, and cannot drive anything the venue does not contain.
- **Strobes are never fired from an AI answer**, even if it asks: DreamShow removes the lighting and effects planned on them, and warns you. Set them yourself if the show requires it.
- Special devices (smoke, fans, UV, machines…) are never switched on by a mood; if the AI switches them on explicitly, a warning asks you to check.
- Nothing is created as long as an error remains.
- The Robot blocks created only make the robot move if it is armed by the control PC (see Humanoid robots).

## Step by step

### Creating the show for a play

1. Check that the right venue is active and that its groups have meaningful names (“Fronts”, “Backlights”, “Moving heads”…): the AI uses them.
2. Open **File > Create a show with an AI…**.
3. Type the **Show name**.
4. Add the sounds: **Add a folder…** and choose the folder of lines and music, or drag the files onto the window. Check that their lengths are displayed.
5. Click **Copy the request**.
6. In your browser, open a conversation with the AI of your choice, paste the request, then attach your script (Word or PDF file) or paste it after it. Send.
7. When the AI has answered, copy **all** of its answer (many AIs offer a “Copy” button under the code block).
8. Go back to DreamShow and click **Paste**: the check starts.
9. If there are errors, copy the red lines, paste them back into the conversation asking the AI to correct them and send the complete answer again, then start again at step 8.
10. When the report says “Everything is understood: you can create the show.”, click **Create the show** ✨.
11. Go through the show scene by scene: adjust the levels, the fades and the cue messages, add what the AI could not know. Save (Ctrl+S).

### Adding sounds afterwards

If the AI asks for a file you had forgotten: add it in part 1 (the show name is already fixed, the file joins the same folder), then click **Check** again. There is no need to copy the request again, unless you want the AI to know the length of the new file.

## Good to know

- The more precise the script is about lighting changes (“night falls”, “a spot on the king at stage left”), the more accurate the show. You can add your own instructions after DreamShow's request.
- The show name is frozen as soon as the media folder exists. To change the name, close the window and reopen it: a new folder will be created.
- The request describes the venue as it is at the moment of copying. If you modify the venue (groups, robots, masks), copy a new request.
- The show created is a complete but simple starting point: one mood per scene. Moving-head movements, trajectories, videos and other blocks are then added in the timeline.

## See also

- Creating your first show
- DMX stream by an AI
- The scene timeline
- Scene settings
- Dynamic effects
- Humanoid robots
- LED masks
- Shows, files and projects

# DMX stream by an AI

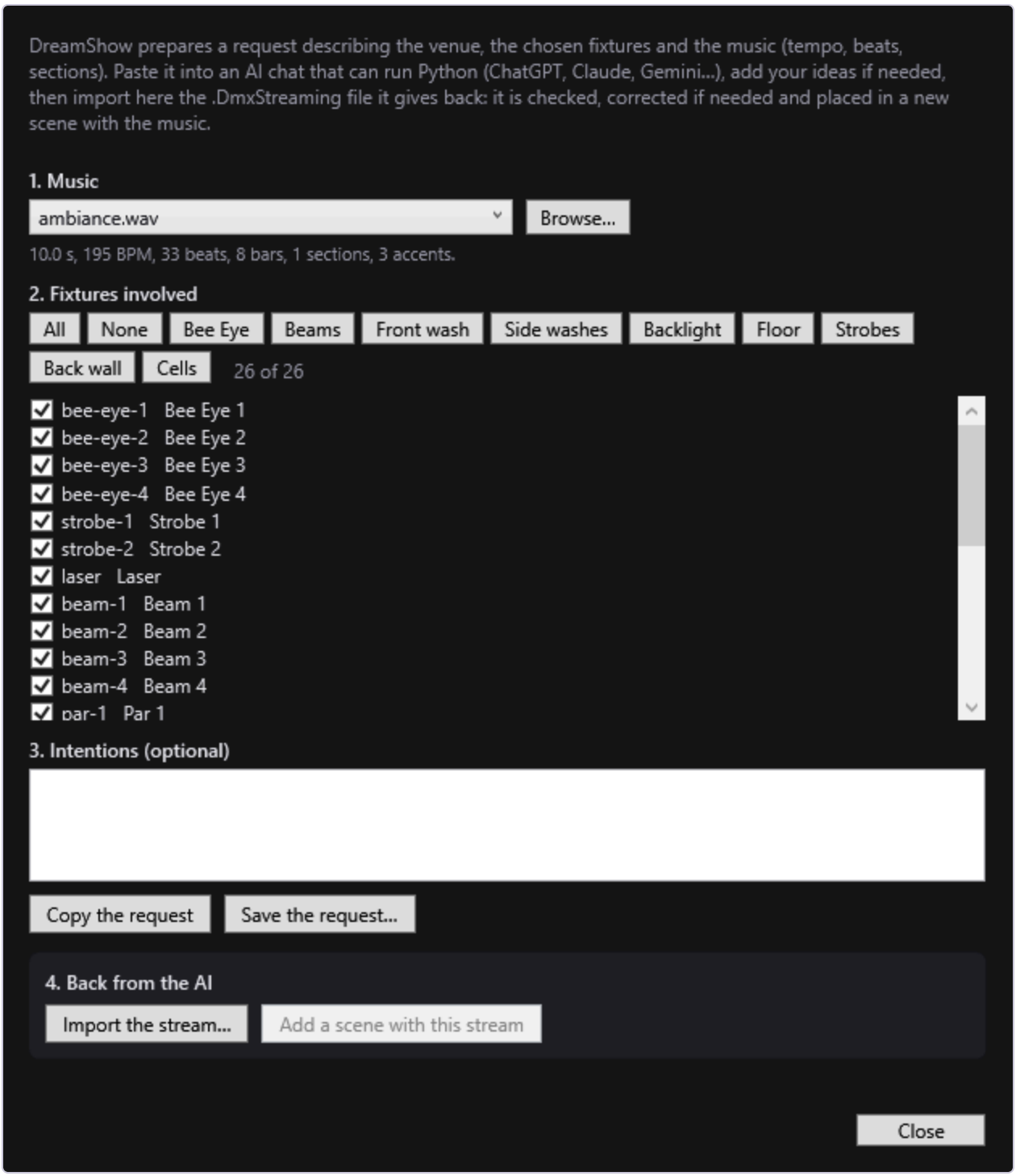
This function has an AI write the complete lighting of a piece of music, frame by frame, as a DMX stream (a `.DmxStreaming` file). DreamShow prepares a request that describes the venue, the chosen fixtures channel by channel, and the analysis of the music (tempo, beats, bars, sections, accents). You paste it into an AI chat able to run Python; the AI gives you a file back, which DreamShow checks, corrects if necessary, then places in a scene with the music.

Unlike Creating a show with an AI, which produces scenes you can edit block by block, the result here is a DMX recording: very rich, perfectly locked to the music, but replayed as it is.

**To open it:** menu **File > DMX stream by an AI…**

The show must have been saved at least once: the stream and the music go into its media folder. Otherwise DreamShow displays “Save the show first: the stream and the music go into its media folder.”. A venue must also be loaded.

The window stays open while you work in the AI chat.




## 1. Music

- The drop-down list offers the sound files already in the show's media folder (wav, mp3, m4a, aac, wma, flac). The first is chosen by default.
- **Browse...**: chooses a sound file elsewhere on the disk; it is added to the list and becomes the chosen music.

As soon as a music file is chosen, DreamShow analyses it (“Analysing the music…”) then displays the result, for example: “214.3 s, 124 BPM, 442 beats, 110 bars, 7 sections, 26 accents.”. During the analysis, the request buttons are greyed out. If the file cannot be read: “Cannot read: …”.

## 2. Fixtures involved

The list shows all the fixtures of the venue, except those marked “invisible”, in the form `identifier name`. All are ticked at the start.

- **All** / **None**: ticks or unticks the whole list.
- One button per **group** of the venue (tooltip “Add or remove this group”): ticks all the fixtures of the group; if they are already all ticked, unticks them.
- The counter “… of …” shows how many are ticked.

Only the ticked fixtures are described to the AI, and only their channels will be accepted on the way back. Untick the devices the music must not drive (smoke, house lighting…).

A fixture whose definition cannot be found, or whose channels go past address 512 of its universe, is set aside; the status line flags it: “Skipped (definition not found or address outside the universe): ….”.

## 3. Intentions (optional)

Free text area: describe what you want (“cold blue mood at the start, an explosion of colours at the chorus, moving heads fanning out on the bridge…”). The text is inserted as it is into the request. Left empty, the request asks the AI for a lighting design that follows the structure of the song, with varied looks and original moving-head movements.

- **Copy the request**: builds the request and copies it to the clipboard. The status line shows its size (“Request: … characters, … fixtures, … channels.”) then “Copied: paste it into the AI chat. If it refuses such a long message, save it and attach the file.”
- **Save the request…**: saves the same request to a Markdown (`.md`) or text (`.txt`) file. Suggested name: `<music name>-ia-consigne.md`. Handy when the chat refuses a message that is too long: attach the file instead.

If no music has been analysed: “Choose the music first.”. If no fixture is ticked: “No fixture selected.”.

### What the request contains

- The request itself: create the lighting of the piece in the form of a `.DmxStreaming` file, with a **required method**: the AI must not write the file by hand, but save the Python module supplied (`dreamshow_stream.py`, without modifying it), write a script that describes the lighting frame by frame, run it, and give you the file to download with its summary. If it cannot run code, it must give you the complete script.
- The name of the file expected: `<music name>-ia.DmxStreaming` (simplified name: lower case, no accents, hyphens).
- Your intentions.
- **The music**: name, length, tempo, number of beats and bars, end of the audible part, accents, and a table of the sections detected with their level and register.

- **The venue**: its dimensions, the reference frame in metres (x from stage right towards stage left, y from the upstage wall towards the audience, z from the floor) and the position of the stage.
- **The fixtures**: for each type, the model, the mode, the number, the moving heads' pan/tilt range, the cells, and a table of **all the channels**: role, neutral value, possible values, **locked** channels and **forbidden** ranges.
- **The rules**: cover the whole piece starting and ending in blackout; never touch locked channels or write forbidden ranges; limited strobe; limited movements; be inventive.
- The file format and the complete **Python module**, with an example script.

**Locked channels and forbidden ranges.** In each fixture's definition, DreamShow finds whatever could make it drop out of the stream or upset it: reset, automatic or sound-driven programs, master/slave modes, lamp, inversion, calibration, display, fan settings… A whole maintenance channel is **locked** (it always stays at its neutral value); elsewhere, only the dangerous ranges are **forbidden**. The neutral value leaves the channel harmless: shutter open, pan and tilt centred, movement speed at maximum, no effect, colour wheel on white.

## 4. The AI's answer

- **Import the stream…**: choose the `.DmxStreaming` file the AI gave you. DreamShow checks it ("Checking the stream…") and writes a **corrected copy** in the show's media folder, under the name of the imported file (with "-2", "-3"… if that name already exists). Your original file is not modified.
- **Add a scene with this stream**: active after a successful import. Adds to the show a scene named "AI: <music name>", placed just after the scene selected in the **Scenes** panel (or at the end if none is selected). It contains:
  - a track "DMX IA" (AI DMX) with a DMX stream block (▦ followed by the file name), of the length of the stream;
  - a **Sound** track with the music, of the length of the piece. If the music came from elsewhere (**Browse…** button), it is copied into the show's media folder. The status line confirms: "Scene "…" added to the show.". The button is then greyed out (one import = one scene).
- **Close**: closes the window.

The fixtures ticked at the time of the import are the ones that count: keep the same selection as when you made the request.

### Checks and corrections

The report is displayed under the buttons. Each line starts with a symbol: ✕ error, ⚠ warning, • remark, ✓ "No problem found.". One line summarises the file: "… frames, … s at … frames/s, … channels.".

**Errors (nothing is imported):**

- “Unreadable file: …”: it is not a valid `.DmxStreaming` file.
- “The file contains no frame.”
- “No channel of the file belongs to the chosen fixtures (wrong venue or wrong patch?).”

**Corrections made in the copy:**

- **Foreign channels**: “… channels outside the selection were removed: they would have driven other devices.”
- **Forgotten channels**: “… channels missing from the file were added at their neutral value.”
- **Forbidden values**: any value written on a locked channel, or in a forbidden range, is set back to neutral: “… forbidden or locked-channel values set back to neutral (…).”, with the five channels most affected.
- **Moving-head speed**: a real moving head does not follow more than about 180° per second on each axis. If the stream goes faster, DreamShow flags it (“Fastest movement: … °/s (…), beyond what a moving head can follow (~180 °/s): the copy slows it down as the fixture would.”) and limits each axis to that speed: the simulation thus shows what the moving head will really do.

**Flags (the copy is not modified):**

- “Continuous strobe for … s (advised: 4 s at most).”
- “Strobe during … % of the song (advised: 10 % at most).”
- “Flashing at … flashes per second (…), above 8.” (flashing done with the intensity).
- “Length … s for music lasting … s.” when the stream and the music differ by more than half a second.

**Remarks:**

- “… chosen fixtures are not in the file: the scene keeps driving them.” (they are not forced to neutral, which would turn them off).
- “File frame rate: … frames per second.” when the stream is not at 40 frames per second.

The strobe flags are comfort and audience-safety alerts: if one of them appears, ask the AI to reduce it, or correct the scene.

## How the stream plays

- While the red line crosses the stream block, each frame of the file is applied at the exact moment. The stream only drives its own channels, on all the universes concerned; the rest of the scene carries on normally.
- Pause freezes the frame; a click on the timeline ruler repositions the stream. The block has no fade: only **Start (s)** and **Length (s)** can be set in its properties (double-click the block).

- The music on the Sound track starts at the same moment: the stream stays in sync.

## Step by step

1. Save the show (Ctrl+S) if it is new, and copy the music into its media folder (**Import media…** button of the **Libraries** panel), or keep it to hand.
2. Open **File > DMX stream by an AI…**.
3. Choose the music and wait for the analysis line.
4. Tick the fixtures involved (the group buttons are quick).
5. Write your intentions, if you have any.
6. Click **Copy the request** (or **Save the request…** if the message is too long for the chat).
7. In an AI chat able to run Python, paste the request (or attach the file) and send. Let the AI write and run its script.
8. Download the `.DmxStreaming` file it produces.
9. In DreamShow, click **Import the stream…** and choose that file. Read the report.
10. In the **Scenes** panel, select the scene after which the new one should come, then click **Add a scene with this stream**.
11. Close the window, select the "AI: …" scene and start playback to see it in the simulation.

If the AI chat cannot run code, it gives you a Python script: run it on your PC with Python 3, in the same folder as the `dreamshow_stream.py` module it contains, to get the file.

## Good to know

- The stream is tied to the venue and to the DMX addresses of the moment: if you re-patch the fixtures, make a new request and a new stream.
- The moving-head movements are calculated for the real position and hanging of each moving head as declared in the venue: place them correctly in **Venue > Fixtures, addresses, positions…** before making the request.
- The request can be long (several tens of thousands of characters with many fixtures): this is normal.
- A `.DmxStreaming` stream is also used by the standalone player.

## See also

- Creating a show with an AI
- The scene timeline
- Sound and video on the timeline
- Fixtures, addresses, positions
- The standalone player
- Shows, files and projects

# AI assistant (set the scene)

Describe the light you want in a few words, for example “a full blue wash covering the right half of the stage”: the assistant sets the selected scene with your real fixtures. It uses Claude, Anthropic's AI, and DreamShow's own tools: covering an area with the moving heads, aiming at a point, colours, intensities, zoom, strobe, palettes. It never sees a DMX address and never touches maintenance channels (reset, automatic programs, sound mode).

**To open it:** menu **File > AI assistant (set the scene)…**. The assistant requires an Internet connection.

## Before the first request: your API key

The assistant works with **your own Anthropic API key**:

1. Create an account on **console.anthropic.com**, then a key in the **API Keys** section.
2. In the assistant, click **API key…**, paste the key and click **Save**.

Usage is billed by Anthropic on your account, a few cents per request. After each request, the bottom of the window shows the tokens sent (including those served from the cache, which cost less) and received.

The key is encrypted by Windows for your user account: it never appears in clear in the settings, and it is only sent to Anthropic. **Delete the key** removes it.

## Making a request

1. Select a scene in the scene list. The top of the window shows “Scene: …”.
2. Type your request and press **Enter** (Shift+Enter for a new line), or click **Send**.
3. The changes show at once in the simulation and in the scene's final state.
4. Click **Keep** to validate them, or **Undo the changes** to put the scene back exactly as it was before the assistant.

Examples:

- “A full blue wash covering the right half of the stage.”
- “The moving heads in amber on the downstage area, at 60 %.”
- “All the moving heads on centre stage, tight beam, at face height.”
- “Add a fast strobe on the front moving heads.” then “Stop the strobe.”
- “Turn off the backlights.”

The conversation keeps track: after “a full blue wash stage right”, you can say “a bit brighter” or “the same on the other side”. **New conversation** starts again from scratch.

## What the assistant understands

- **Right and left** are **as seen from the audience**: right = “cour”, left = “jardin”. For the opposite, say “actor's right”. The assistant says in its answer which side it took.
- **Areas**: it knows the stage width and depth and the position of every fixture. To cover an area, it spreads the **moving heads** over it and adjusts their zoom. Fixed fixtures do not move: it picks those that already light that area.
- **Names**: it only uses the venue's group and fixture names. An unknown name is refused without changing anything.
- **Palettes**: it can apply the venue's colour, position or beam palettes.
- When a request is ambiguous, it asks a question instead of acting.

## What it does not do (yet)

The assistant sets **one scene at a time**. It does not create scenes, does not touch the timeline (blocks, fades, dynamic effects, sounds, videos) and does not change an effect being edited: close the effect to use it.

## If something goes wrong

- “API key rejected by Anthropic”: the key is wrong or was deleted; paste it again with **API key…**.
- “No Internet connection”: the assistant needs Internet; the rest of DreamShow works without it.
- “Too many requests for this API key”: wait a minute.
- On any error, the scene is put back as it was before the request.

See also: Creating a show with an AI, Palettes, Selection and light settings.

# Connect an AI (MCP)

The AI you already use can set DreamShow's scenes when you ask it: “full blue wash stage right”, “switch off the upstage moving heads”… It works with Claude, ChatGPT, Claude Code, Cursor, VS Code (Copilot), Windsurf, LM Studio, Gemini CLI and any other MCP-compatible AI (the open standard that lets an AI use a program). No API key is needed: your AI, with its own subscription, does the thinking.

**To open it:** menu **File > Connect an AI (MCP)…**.

## Connect your AI in three steps

1. **Choose your AI** in the list. AIs installed on this PC are marked.
2. **Quit your AI completely** (Claude, ChatGPT: right-click its icon near the clock, then Quit), then **click “Connect DreamShow to …”**. The button waits until the AI is closed: otherwise it would erase the connection when closing. DreamShow adds itself to your AI's configuration (a copy of the old configuration is kept next to it, with the extension `.avant-dreamshow`).
3. **Reopen your AI** as shown, then type the suggested sentence: “Which scenes are there in DreamShow?”. As soon as it answers, the window shows **“It works!”**.

DreamShow must stay open while the AI works. The **Disconnect** button removes DreamShow from the AI's configuration.

## Working with the AI

Open the venue and show you want, then just ask, for example: “In scene 2, put a full blue wash on the stage-right half of the stage.” The result shows at once in the simulation and in the status bar (“AI (MCP): …”).

- Right and left are **as seen from the audience** (right = “cour”).
- The AI's changes **can be undone**: in the window, **Keep** validates them and **Undo the changes** puts the scene back as it was. You can also tell the AI “undo what you just did”.
- The AI has the same safe tools as the built-in assistant: areas, aiming, colours, intensities, zoom, strobe, palettes. It never sees a DMX address.

## ChatGPT

With the ChatGPT app for Windows, DreamShow is used by Codex. If an ordinary conversation does not see DreamShow, ask the question in Codex.

## AIs in a web browser

claude.ai, ChatGPT on the web and other AIs used in a browser cannot reach software installed on your PC: use their app, or DreamShow's built-in AI assistant.

## Advanced settings

For an AI that is not in the list, **Advanced settings…** gives the configuration blocks to copy (started by DreamShow, or the address `http://127.0.0.1:8097/mcp` with a token). The connector only listens on this PC and requires the token; **New token** creates another one and updates the AIs connected by the wizard.

## If something goes wrong

- The AI answers that “DreamShow is not open, or its MCP connector is not enabled”: open DreamShow; the connector comes back on by itself.
- The window stays on “Waiting for a first exchange”: check that the AI was fully restarted.
- “The MCP connector could not open port 8097”: another program uses that port; close it or contact support.

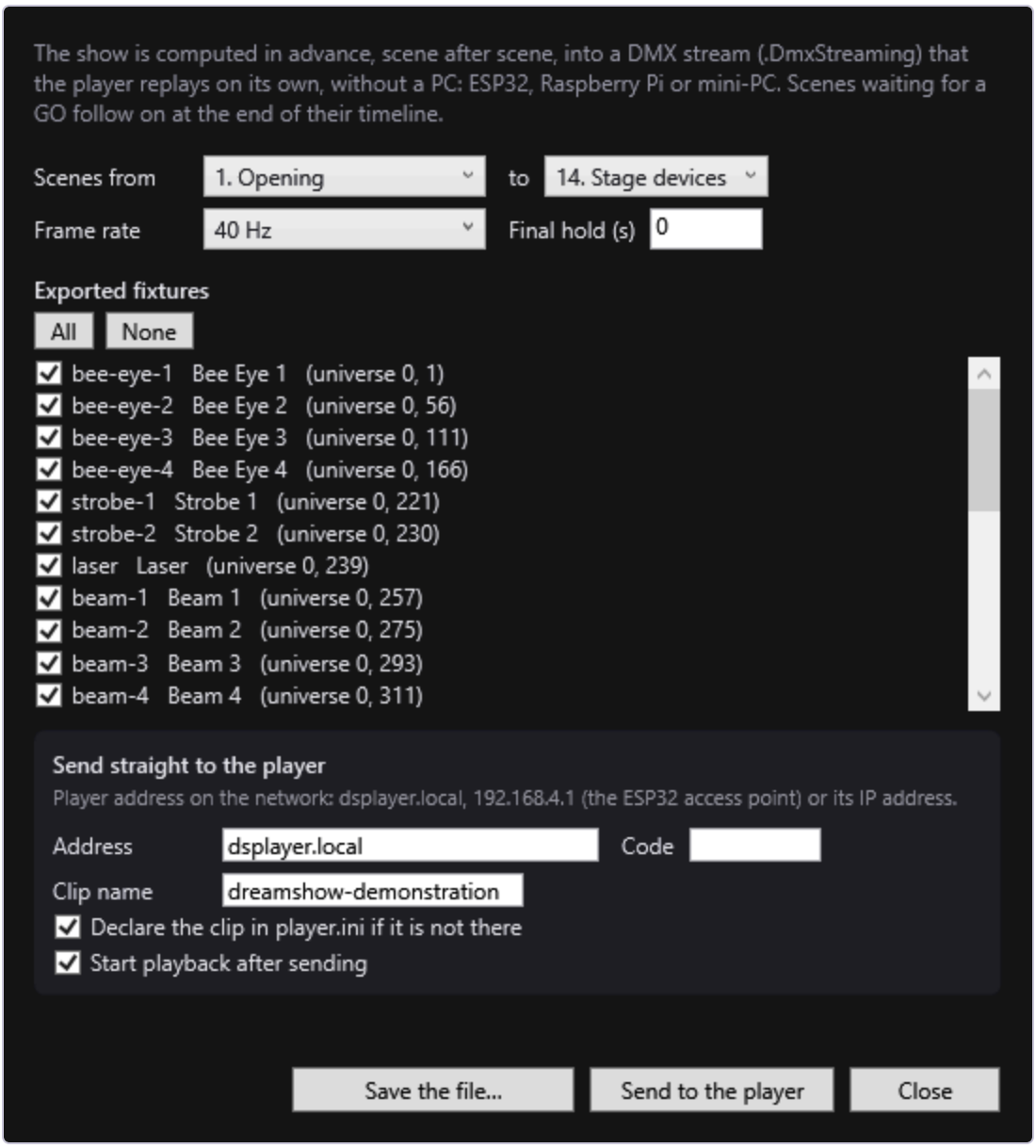
See also: AI assistant (set the scene), Creating a show with an AI.

# The standalone player

The standalone player replays a DreamShow show **without a PC**: a façade, a shop window, a lobby, an illuminated garden, a carousel… DreamShow calculates the show in advance, scene after scene, into a DMX stream (a `.DmxStreaming` file), which you send to a small box: an ESP32 module, a Raspberry Pi or a Windows mini-PC. The player replays it on its own, following its own timetable, including at sunset.

**To open it:** menu **File > Export for a standalone player…** (tooltip: “DMX stream (.DmxStreaming) replayed without a PC by an ESP32, a Raspberry Pi or a mini-PC”).

Conditions: a venue loaded and a show containing at least one scene (otherwise “The show contains no scene.”).




## How the show is calculated

As the top of the window explains: “The show is computed in advance, scene after scene, into a DMX stream (.DmxStreaming) that the player replays on its own, without a PC: ESP32, Raspberry Pi or mini-PC. Scenes waiting for a GO follow on at the end of their timeline.”

More precisely, DreamShow replays the show with its own engine, as fast as the processor allows:

- scenes set to **Auto** or **Chained** move on to the next one as they would in performance;
- a scene set to **Manual GO** receives its GO automatically when its timeline is finished (plus the hold you set, see below);
- a scene whose timeline is empty lasts at least 2 seconds;
- the fades between scenes, effects, trajectories, dynamic effects, pixel mapping and DMX streams are calculated as in DreamShow;

- only the **light** is exported: sound, video, robots, masks, displays and the other devices are not in the stream;
- waits for a time, a date or the sun are skipped (the player has its own schedule, see below);
- a scene set to **Loop** with a **max length** is exported up to that length; without a max length, a single turn is exported, and it is the player that replays the clip in a loop;
- a **Go to scene** jump to a scene that has already been played ends the export: a single cycle is calculated, which the player replays in a loop from the start of the clip;
- as a safety measure, the calculation stops after one hour of show.

## The settings

- **Scenes from … to …**: the first and last scenes exported (by default: the whole show). Scenes are listed in the form "3. Scene name".
- **Frame rate**: number of frames per second of the stream: **25 Hz**, **30 Hz** or **40 Hz** (40 Hz by default, the frame rate of the DreamShow engine). A lower frame rate gives a smaller file, useful on an ESP32.
- **Final hold (s)**: number of seconds added at the end of the timeline of a Manual GO scene before its automatic GO (0 by default). It applies to **every** exported Manual GO scene, the last one included: this is how you keep a scene's final state displayed for a while.

### Exported fixtures

The list shows all the fixtures of the venue, in the form `identifier name (universe U, address)`, all ticked at the start.

- **All** / **None**: ticks or unticks everything.

The stream contains all the channels of the ticked fixtures, and only those. Untick the ones the player must not drive. If none is ticked: "No fixture selected.".

## Saving the file

- **Save the file…**: suggests a name (`<clip name>.DmxStreaming`), calculates the stream ("Computing the stream…") and saves it. The summary is displayed: "Saved: …" then "… s, … scenes, … fixtures, … channels on … universes, … KB.".

You can then copy this file to the SD card of a Raspberry Pi, to the disk of a mini-PC, or upload it from the player's web page.

If the file exceeds about 2.4 MB, the summary adds in orange: "Too large for the memory of a 4 MB ESP32: use an SD card, a 16 MB module or a Raspberry Pi."

## Send straight to the player

The **Send straight to the player** frame sends the stream over the network, without going through a file.

- **Address**: the address of the player. As the window says: “dsplayer.local, 192.168.4.1 (the ESP32 access point) or its IP address.” Default value: `dsplayer.local`. DreamShow remembers the last address used.
- **Code**: the player's code, if it has one (the player's `pin` setting). Leave empty otherwise.
- **Clip name**: the name under which the player will know this show. Suggested from the show's name (lower case, no accents, hyphens, 40 characters at most); the file sent is called `<clip name>.DmxStreaming`.
- **Declare the clip in player.ini if it is not there** (ticked by default): adds the line `<clip name> = <file>` to the `[clips]` section of the player's configuration, if no clip of that name appears there. The player then reloads its configuration.
- **Start playback after sending** (ticked by default; DreamShow remembers your choice): asks the player to play at once the clip that reads this file. This option requires the clip to be declared in `player.ini`; otherwise: “No clip in player.ini plays …: tick the clip declaration.”.
- **Send to the player**: calculates the stream, sends it (“Sending … KB to …”), then displays “Sent to player …: ….”, possibly “Clip "…" added to player.ini.” and “Playback of "…" started.”, followed by the summary.

In case of failure: “Sending failed: …”, with the reason, for example “no answer from the player”, “The player refused the file (…).” (the number 403 in brackets means that the code is missing or wrong), “The player refused player.ini (…).” or “The player refused to play (…).”.

A clip sent again while it is playing carries on with the new version.

- **Close**: closes the window.

## The player itself

The player is a program supplied with DreamShow for three kinds of boxes:

| Box | DMX outputs | Buttons / relays | Clip storage |
|---|---|---|---|
| ESP32 (ESP32, ESP32-S3, ESP32-C3) | serial port + RS-485 module, Art-Net over Wi-Fi | GPIO pins | internal memory (about 2.4 MB on a 4 MB module, 14 MB on a 16 MB module) or SD card |
| Raspberry Pi | serial port + RS-485, “Open DMX” FTDI cable, Enttec USB PRO type interface, Art-Net | GPIO pins | the Pi's SD card |

| | | | |
|---|---|---|---|
| Windows PC (mini-PC) | Enttec USB PRO type interface, Art-Net | keyboard keys | disk |

What it can do:

- several **clips** (a clip = a `.DmxStreaming` file);
- a **weekly schedule**: time slots by day of the week, at a fixed time or at **sunrise / sunset** with an offset in minutes, including slots that run past midnight;
- playback in a **loop** or just once, **crossfade** between clips, and, outside the slots, **blackout** or holding the last frame;
- **buttons** (GO, next, previous, blackout, back to the schedule) and **relays** driven according to the state (playing, stopped, a given clip);
- several universes: each universe of the stream goes to a DMX port or an Art-Net universe;
- a **web page** for control.

### The player's web page

Open the player's address in a browser (for example `http://dsplayer.local/`). The page is in French if the browser is, in English otherwise. It shows:

- the player's name, the time, the sunrise and sunset times and the version;
- the mode (**Schedule** or **Manual**), the current slot, the clip playing, a progress bar and the position;
- the buttons **GO**, **◀ Previous**, **Next ▶**, **BLACKOUT**, **Stop** and **Schedule** (back to the timetable);
- the list of **Clips**, with a ▶ button to play one, and the mention “missing” if the file is absent; a file field and the **Upload** button to send a clip;
- the editable **Configuration (player.ini)**, with **Save and reload**, and a **code:** field for the player's code.

Without Internet, the player takes the time from the browser when the page is opened; with Internet, it takes it from the network.

### The player.ini file

The player's configuration is held in a text file, editable from the web page. Example:

```
[player]
name      = Facade Paradise
latitude  = 44.126
longitude = 4.082
idle      = black          ; hors planning : black (noir) ou hold (maintien)
fade      = 2.0            ; fondu entre clips, en secondes
; pin     = 1234           ; code demandé pour modifier (page web, envoi)

[clips]
```

```
soir = soir.DmxStreaming
nuit = nuit.DmxStreaming

[schedule]
; nom = jours(lundi..dimanche) début fin clip [loop|once]
soiree = 1111111 sunset-15 23:30 soir loop
nuit   = 0000011 23:30 01:00 nuit loop
```

- The **days** are written with seven digits, from Monday to Sunday (1 = active).
- The **start** and **end** are written `HH:MM`, `sunrise` or `sunset`, with an optional offset in minutes (`sunset-15` = a quarter of an hour before sunset).
- The first active slot wins.
- Latitude and longitude are used to calculate the sun.

The `[wifi]` (ESP32), `[output]` (DMX and Art-Net ports), `[buttons]` and `[relays]` sections set up the hardware; their detail is given in the player's documentation.

### First start of an ESP32

With no Wi-Fi network configured, the ESP32 opens its own access point **DreamShow-Player** (password `dreamshow`); its page is then at `http://192.168.4.1/`. Fill in the `[wifi]` section there and save: the player joins your network and then answers at `http://dsplayer.local/`.

## Step by step

### Lighting a façade every evening without a PC

1. Prepare the show in DreamShow: one or more scenes set to **Loop** or chained, checked in the simulation.
2. Open **File > Export for a standalone player...**.
3. Choose the scenes and leave the frame rate at 40 Hz (25 Hz if the file is too large for an ESP32).
4. Untick the devices the player does not drive.
5. In **Send straight to the player**, enter the player's address, its code if it has one, and a clip name, for example `soir`.
6. Leave both boxes ticked, then click **Send to the player**. The player plays the clip at once.
7. Open the player's web page, and in **Configuration (player.ini)**, add the slot you want in `[schedule]`, for example `soiree = 1111111 sunset-15 23:30 soir loop`, then **Save and reload**.
8. Click **Schedule**: the player now follows its timetable.

### Preparing a clip for an SD card

1. Open the window, set the scenes and the fixtures.

2. Click **Save the file…** and choose the SD card or a USB stick.
3. Declare the file in the `[clips]` section of the box's `player.ini`.

## Good to know

- The timeline's **waits for a time, a date or the sun** do not go into the stream: on the player, it is its **schedule** that decides when to play each clip. Make one clip per moment of the day.
- Sound and video are not played by the player.
- Calculation and sending are done in the background; the window stays usable, but the buttons are greyed out while it works.
- The exported stream has the same format as the one in DMX stream by an AI.

## See also

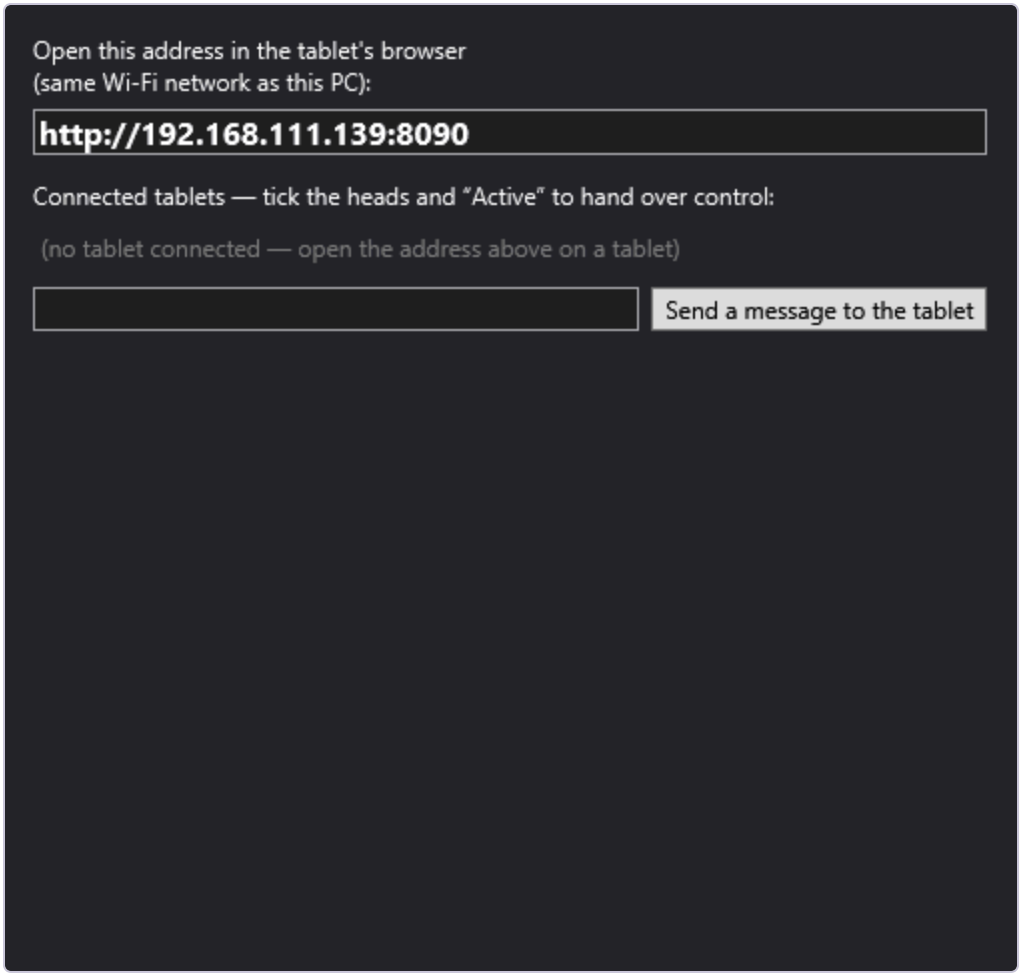
- Architectural lighting
- Waits (key, time, date, sun)
- Scene settings
- DMX connections
- DMX stream by an AI

# The follow spot on a tablet

One or more tablets (or phones) serve as a “follow spot”: on the screen, a plan of the venue; you put your finger down, and the chosen moving heads follow your finger live. Each tablet can drive its own moving heads, with its own intensity and zoom. It is ideal for following an actor, a singer or a presenter without a traditional follow spot.

**To open it:** menu **Venue > Follow spot (tablets)…** (tooltip: “Drive one or more moving heads with a finger from a tablet (browser, same Wi-Fi network) — follow spot.”).

There is nothing to install on the tablet: it uses its web browser. The tablet and the control PC must be on the same network (the same Wi-Fi, or a Wi-Fi connected to the PC's network).




## What happens when it opens

The first time the window is opened, DreamShow starts a small web server on **port 8090** of the PC. The log says “Server started: http://<PC address>:8090”. The server then stays running until

DreamShow is closed, even if you close the window.

## The Follow spot (tablets) window

### The address to open

“Open this address in the tablet's browser (same Wi-Fi network as this PC):” followed by the address, in bold, for example `http://192.168.1.20:8090`. The text can be selected and copied.

DreamShow shows the first network address (IPv4) of the PC. If the PC has several network cards, it may not be the one on the tablets' network: in that case use the PC's address on that network, with `:8090`.

### The connected tablets

“Connected tablets — tick the heads and “Active” to hand over control:”

Each tablet appears in a frame, identified by its IP address. If none has connected: “(no tablet connected — open the address above on a tablet)”.

In each frame:

- **The state** of the tablet, in colour:
  - ● green “IN CONTROL”: the tablet is active and has moving heads; “· pilotage en cours” (“driving in progress”) is added when it has sent an aim in the last two seconds;
  - ● orange “active — no head assigned”;
  - ● grey “connected (inactive)”;
  - ○ dark grey “disconnected”: the tablet has lost the link, but its settings are kept.
- **Active**: hands control to this tablet. On activation, DreamShow reads where the first ticked moving head is already pointing and places the tablet's point there: the moving head does not jump. Unticking gives the moving heads back to the show.
- **One box per moving head** of the venue: the moving heads this tablet drives. Only the moving heads whose DMX mode really has pan/tilt channels are offered. “(not cal.)” flags a moving head with no calibration: it aims by calculation, from its position declared in the venue. If the venue has no steerable moving head: “(no moving head in this venue)”.
- **Last aim**: “last aim: u=… v=… · intensity … %”, that is the last position of the finger on the plan (from 0 to 1 in width and in depth) and the intensity. For a disconnected tablet, “(link closed — assignment kept for reconnection)” is added.

A tablet is recognised by its IP address: if it reconnects (standby, Wi-Fi dropped), it gets back its moving heads and its active state.

### Sending a message to the tablets

At the bottom: a text area and the **Send a message to the tablet** button. The text is displayed at the top of the screen of **all** the connected tablets (“get ready for the king's entrance”, “follow spot on the singer”…). Send an empty message to clear it.

## The tablet screen

The tablet page itself is displayed in French (its labels are quoted below with their translation).

- **The plan**: the venue seen from above, **unfolded**: the floor in the centre, the stage drawn on it (“scène”, the stage), and the four walls folded out all around, as in DreamShow's plan (“mur du fond”, the back wall, at the top, “mur avant”, the front wall, at the bottom, “jardin”, stage right, on the left, “cour”, stage left, on the right). The plan keeps the proportions of the venue.
- **The yellow dot**: where the moving heads are aiming. Put your finger down and drag it: the moving heads follow. Dragging onto a strip of wall makes the beam climb up that wall.
- **Int.** (intensity): vertical slider for the intensity of the driven moving heads (100 % at the start).
- **Zoom**: vertical zoom slider, from the tightest (bottom) to the widest (top); no effect on a moving head without a zoom channel.
- **The status line** at the bottom left: “connecté” (connected), “déconnecté — reconnexion…” (disconnected — reconnecting…; the tablet reconnects by itself every second), the position of the finger, or “position lyre : …” (moving head position: …) when the control PC has reset the dot.

## How the moving heads are driven

- On the stage and on the floor, a **calibrated** moving head uses its calibration (the most accurate); elsewhere, or without calibration, DreamShow calculates the aim from the position, orientation and hanging direction of the moving head declared in the venue.
- For a horizontal sweep, DreamShow always starts from the moving head's current position: it does not make a needless full turn.
- The moving heads driven by an active tablet **override the show**: pan, tilt, intensity and zoom come from the tablet, whatever scene is playing. When you untick **Active** or the moving head, it goes back to the cue list.
- Several tablets can be active at the same time, each with its own moving heads. Avoid giving the same moving head to two active tablets.

## Step by step

1. Preferably calibrate the follow-spot moving heads (see Moving-head calibration) and check their position in **Venue > Fixtures, addresses, positions…**.
2. Open **Venue > Follow spot (tablets)…**.
3. On the tablet, connected to the same Wi-Fi, open the browser and type the address shown. The tablet's status line says “connecté”; in DreamShow, a frame appears with its IP address.

4. In this frame, tick the follow-spot moving head(s).
5. At the right moment, tick **Active**: the state changes to “IN CONTROL”.
6. On the tablet, put your finger on the actor (on the plan) and follow them. Adjust **Int.** and **Zoom**.
7. At the end, untick **Active**: the moving heads go back to the cue list.

## Good to know

- **The BLACKOUT does not cut the follow spot**: the moving heads of an active tablet override everything, general blackout included. To turn them off, lower **Int.** on the tablet or untick **Active**.
- Closing the window does **not** give control back: the server and the active tablets carry on. Untick **Active** before closing.
- The connection has no password: reserve it for a trusted network.
- When the server first starts, the Windows firewall may ask you to allow DreamShow on the network: accept, otherwise the tablets will not be able to connect.
- There is no “Follow spot” timeline block: control is handed over and taken back in this window.

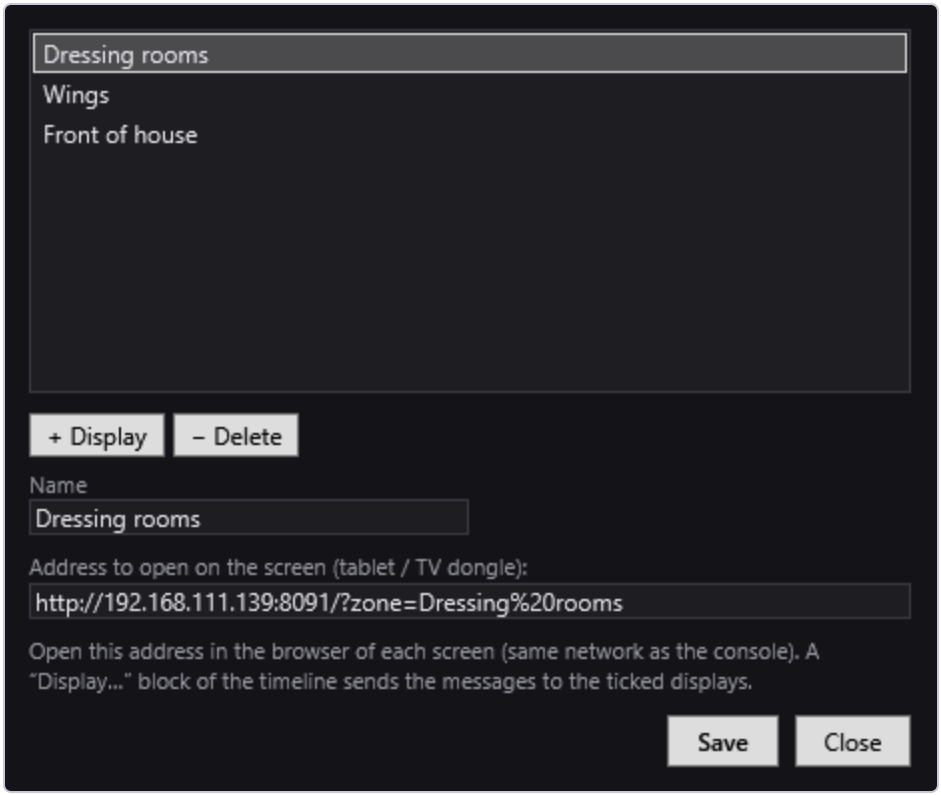
## See also

- Moving-head calibration
- Aiming and covering a zone with moving heads
- Manual controls
- The web remote
- Troubleshooting

# Backstage displays

Backstage displays are information screens placed in the wings, in the dressing rooms or at the front of house: “5 minutes to curtain up”, “Interval — resuming at 9:15 pm”, “Current scene: Act II”, the time, the length of the show… DreamShow itself serves the page to display; any screen with a browser (tablet, TV with an HDMI stick, mini-PC, phone) shows it full screen. Messages are sent from the timeline, at the right moment of the show, by the **Backstage display…** blocks.

**To open it:** menu **Venue > Backstage displays…** (tooltip: “Information monitors (wings, dressing rooms): a web page served by DreamShow (port 8091). An “Display…” block of the timeline sends messages to it.”).




## The principle

- Each display has a **name** (also called a “zone”): “Dressing rooms”, “Wings stage right”, “Front of house”…
- Each screen opens the address of **its own** display: `http://<address of the control PC>:8091/?zone=<name>`.
- A **Backstage display…** block on the timeline sends a message to one or several displays. The message stays on screen until the next block aimed at the same display. An empty message clears the screen.

The display server starts when the venue declares at least one display, or when this window is opened. The log then says “Display server started: http://…:8091”.

## The Backstage displays window

The title recalls the venue: “Backstage displays: <venue>”. An asterisk * is added to the title as long as changes have not been saved.

- The **list** of the venue's displays.
- **+ Display**: adds a display named “Display” (then “Display 2”, “Display 3”…) and selects it.
- **− Delete**: removes the selected display.
- **Name**: the name of the selected display. This is the name the Backstage display… blocks offer, and the one used in the address. Prefer short names with no special characters.
- **Address to open on the screen (tablet / TV dongle):** the full address of this display, to be copied into the screen's browser. It updates when you change the name. As the window reminds you: “Open this address in the browser of each screen (same network as the console). A “Display…” block of the timeline sends the messages to the ticked displays.”
- **Save**: saves the list in the venue. Displays without a name are removed.
- **Close**: closes the window. If changes remain, DreamShow asks “Display changes not saved. Save?” (Yes / No / Cancel).

The address shown uses the PC's first network address (IPv4). If the PC has several network cards, replace it if needed with its address on the screens' network.

## A display's screen

The page appears on a black background, with nothing to set up:

- in the centre, **the message**, in very large characters, with its line breaks; it fades when it is empty;
- at the bottom, a bar with the display's name on the left (● Dressing rooms), “duration” followed by the length of the show in the middle, and the time on the right, to the second.

The page updates by itself. If the link is cut (PC switched off, Wi-Fi lost), it reconnects by itself about every second and a half. A screen that connects immediately receives the last message sent to its display.

Put the screen's browser in full-screen mode (F11 on a PC, kiosk mode on a tablet).

## LED screen and teleprompter

A display can also be an **LED wall or panel**: a TV dongle (Xiaomi TV Stick, Fire TV…) running a borderless browser, plugged into the HDMI input of the panel's sending card, opens the display's

address. The control PC gives up no video output: everything goes over the network.

A sending card (Novastar MCTRL…) does not rescale the picture: it shows, pixel for pixel, the part of the HDMI picture that matches the size of the panel, often the top-left corner. For a small 192 × 96 panel, only that corner of a 1920 × 1080 picture is visible. Set it in the Backstage displays window, under **LED screen or partial area**, for the selected display:

- **Dongle HDMI picture**: the resolution the dongle sends (for example 1920 × 1080).
- **Visible part: X, Y, Width, Height**: the part of that picture the panel shows, in pixels (empty = up to the edge).
- **Mirror (glass teleprompter)**: flips the text left to right.
- **Hide the bar (zone, duration, time)**: keeps only the message, useful on a small panel.
- **Show the alignment pattern on this screen**: a red frame, its diagonals and its size replace the message. The frame must fall exactly on the edges of the panel. The pattern is not saved; it disappears when the window is closed.

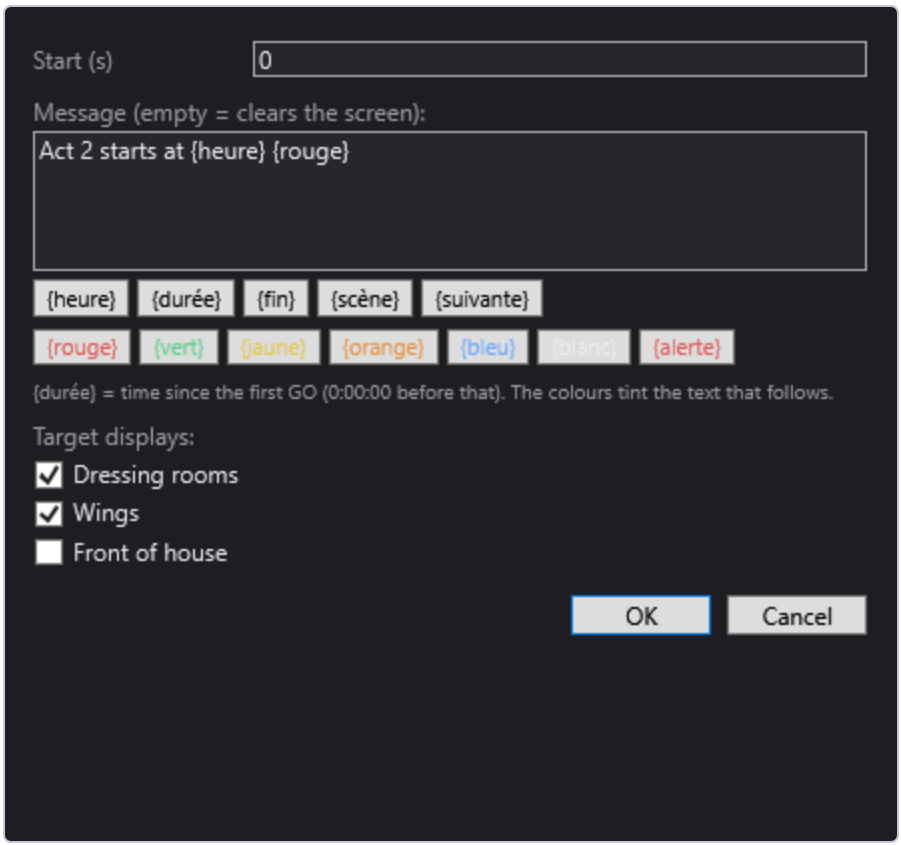
As soon as a visible part is set, the message is written in bold, **as large as the area allows**, without breaking words, and the rest of the picture is black. Changes appear on the screen at once, without reloading it. Under the address, the window tells whether a screen is connected to this display.

The setting is in pixels of the HDMI picture, whatever the scale of the dongle's browser. If the text looks shifted, check that the dongle applies no zoom, overscan or screen-position setting.

## The Backstage display… block

In the **Libraries** panel, the **Others** list contains **Backstage display…**. Drag it onto a track of the timeline, at the point where the message must be sent: the “Backstage display” (Afficheur backstage) window opens at once. If you cancel it, no block is created. To edit an existing block: double-click it, or right-click > **Properties…**.

On the timeline, the block shows 🖵, the beginning of the message and its recipients, for example “🖵 Interval → Dressing rooms, Front of house”. A block with no message shows “(clear)”; with no recipient, “→ (none)”.




### The block's settings

- **Start (s)**: the moment, counted from the start of the scene, when the message is sent.
- **Message (empty = clears the screen):** the text, over several lines if you wish (Enter key for a new line).
- **The token buttons**, which insert a keyword at the cursor; on the screen, each token is replaced by its value and kept up to date:

| Token | Replaced by |
|---|---|
| {heure} | the time, to the second (the screen's clock) |
| {durée} | the time elapsed since the show's first GO, in the form 0:00:00 (0:00:00 before the first GO) |
| {fin} | the time (hh:mm) at which the show reached its last scene, "—" before that |
| {scène} | the current scene, with its number ("4 · Act II") |
| {suivante} | the next scene, with its number |

- **The colour buttons**: **{rouge}** (red), **{vert}** (green), **{jaune}** (yellow), **{orange}** (orange), **{bleu}** (blue), **{blanc}** (white) colour all the text that **follows** them (up to the next colour token). **{alerte}** (alert) turns the whole screen background red (the word itself is not displayed).
- The reminder: "{durée} = time since the first GO (0:00:00 before that). The colours tint the text that follows."

- **Target displays:** one box per display of the venue; tick those that must receive this message. If the venue has none: “No display declared (Venue → Backstage displays…).”.
- **OK** saves the block, **Cancel** abandons it.

A block with no display ticked sends nothing.

### Example messages

- `Curtain up in 5 minutes` → Dressing rooms, Wings.
- `{rouge}SILENCE{blanc}` then, on a new line, `Scene: {scène}` → Wings.
- `{alerte}SHOW STOPPED` → all displays.
- `Start at 8:30 pm — duration {durée}` → Front of house.
- Empty message → clears the ticked screens.

## Step by step

1. Open **Venue > Backstage displays…**.
2. Click **+ Display** and name it “Dressing rooms”. Repeat for “Wings”. Click **Save**.
3. On each screen, connected to the same network as the control PC, open its address (shown under the name when you select it) and switch the browser to full screen. The bottom bar shows “● Dressing rooms” (or “● Wings”).
4. In the pre-show scene, drag **Backstage display…** from **Others** onto a track, at 0 s. Write `Doors open — start in 15 min`, tick “Dressing rooms” and “Wings”, **OK**.
5. In the first scene of the show, add a block `{vert}ON STAGE{blanc}` then, on a new line, `{scène} — {durée}`, for “Wings”.
6. Switch to show mode and press GO: the screens follow the show.

## Good to know

- {heure} and {durée} are computed by the screen using its own clock: set the screens' time if it is wrong.
- {durée} keeps running on the displays even after the last scene has been reached, whereas the cue window stops on the final duration.
- On the page, {duree} and {scene} (without accents) work as well; two colours with no button exist: {cyan} and {gris} (grey).
- Messages are sent when the red line passes over the block, in performance as well as during playback in the editor.
- The server listens on port **8091**; at first start-up, the Windows firewall may ask you to allow DreamShow on the network: accept.
- The page has no password: the screens must be on a trusted network.

## See also

- The scene timeline
- Show mode and the cue window
- The web remote
- Troubleshooting

# The teleprompter

The teleprompter shows the play's text, one line at a time, on a backstage display: a screen in the wings, a tablet, or an LED panel facing the actors. It moves **independently of the cue list**: the person in charge advances it from a **phone**, without touching light or sound; the stage manager can also advance it with keys of the conductor.

**To open it:** menu **File > Teleprompter...**. The text is saved in the show.

## The text

Click **Import a text...** and choose a text file (UTF-8 or ANSI) in the répétiteur format:

- each line starts with `@NAME` followed by the text, or with `NAME:` at the start of a line; the name is in capitals (`JEAN-CLAUDE`, `L'HOMME` are fine);
- a `# title` line marks the start of a scene (`# Act II`); it is not shown on the screen;
- two lines in a row by the same character are joined.

The window sums up the text: number of lines, characters, text scenes, and how many of those scenes carry the name of a scene of the show. Each character gets a colour. Importing again keeps the settings; **Remove** takes the text out of the show.

## The settings

- **Display**: the backstage display that shows the text (declare it first in **Venue > Backstage displays...**). For an LED panel, also set the visible part of the picture there and hide the bar: see Backstage displays.
- **Max characters per screen**: a longer line is cut into several screens, at the end of a sentence if possible, otherwise after a comma, otherwise between two words. 0 = the whole line on one screen. On a small panel, lower this number to keep the text readable.
- **Show the character's name above the text**, in its colour.
- **Follow the show's scenes**: when the show enters a scene, the teleprompter jumps to the first line of the text scene carrying **the same name** (capitals and accents ignored; the scene's short name counts too). So name your DreamShow scenes like the `#` markers of the text. A scene without a matching marker does not move the teleprompter.

The list shows every screen; the one on display is highlighted. Double-click a screen to show it. At the bottom: ◀ **Previous**, **BLACKOUT** (clears the screen; **Show the text again** brings it back) and **Next** ▶, with the position (« Screen 12 / 870 »).

## By voice

Tick **Advance by voice**: the lines microphone (the same as for the Wait for a spoken line blocks, chosen just below) listens to the stage, and the prompter moves to the next screen by itself.

- The screen changes as soon as **all of it but its last word** has been said, or at the first pause once most of it was said: a line said a little differently from the text moves it on too.
- If a passage is skipped, or not heard (a voice coming from a loudspeaker far from the microphone), the prompter catches up as soon as it recognises what follows. If the text is taken up again earlier, it goes back.
- To jump ahead or back, it needs a **consistent passage** of several words: a few words that occur all over the text (a first name, "de", "le") are never enough.
- The hand keeps priority: a move from the phone, the keys or this window puts the listening there. While **BLACK** is on, nothing moves.
- Under the box, the window shows the microphone listened to and the last words heard.

**Show the next screen greyed out** adds the next screen, in grey, below the current one: the actor sees their line coming while the other one speaks. It helps by voice when the previous line is said by a voice the microphone hears badly.

Recognition runs on the PC, without Internet. **Spoken language** (under the box) chooses the language of the text: French, English, German, Spanish or Italian — the same as the show's **Wait for a spoken line…** blocks. French comes with DreamShow; another language is downloaded once (40 to 50 MB) when it is chosen. Place the microphone close to the actors: the more clearly it hears, the better the screen changes on time.

## From a phone

On the phone, on the same network as the control desk, open the address shown in the window: `http://<control PC address>:8091/prompteur`. The page shows the whole text, the line on screen highlighted and centred, the next one slightly brighter:

- **SUIVANT ▶** (next, big button under the thumb), ◀ to go back, **NOIR** to clear the screen (**TEXTE** to bring it back);
- tap a line, then **Afficher ici** to jump to it;
- **A−** / **A+** set the text size on the phone;
- a Bluetooth presentation remote paired with the phone works too: arrows, Page Down/Page Up, Space (next/previous), B or full stop (black).

A green dot at the top shows the link; if it is cut, a red banner appears and the page reconnects by itself. Several phones can be connected: they all follow the same position. Set the phone's sleep delay to a long time; the page cannot prevent it.

## From the conductor

In **File > Remote (GO keys)…**, the sections **Teleprompter: next screen** and **Teleprompter: previous screen** take the keys of your choice (none by default). They act in the conductor window, in show mode.

## Good to know

- The screen and the phone page go through the display server (port 8091), which starts as soon as the show has a text and the venue declares a display.
- A timeline **Backstage display…** block aimed at the same display replaces the text until the next teleprompter move.
- The position is not saved: when the show is reopened, the teleprompter starts again from the beginning (or from the current scene if it follows the scenes).
- The phone page has no password: trusted network only.

## See also

- Backstage displays
- Show mode and the conductor window

# Humanoid robots

DreamShow drives Booster K1 humanoid robots from the timeline: standing up, speaking (with a sound file), dancing, waving, walking, turning the head… Each action is a **Robot…** block placed at the right moment of the scene, synchronised with the light, the sound and, if you wish, the mouth of an LED mask. As a safety measure, **a robot only moves if the control PC has armed it**.

The robot must run the DreamShow service (the `dreamshow_robot.py` program installed on the robot, started at boot). DreamShow sends it simple orders over the network; the robot must therefore be on the same network as the control PC.

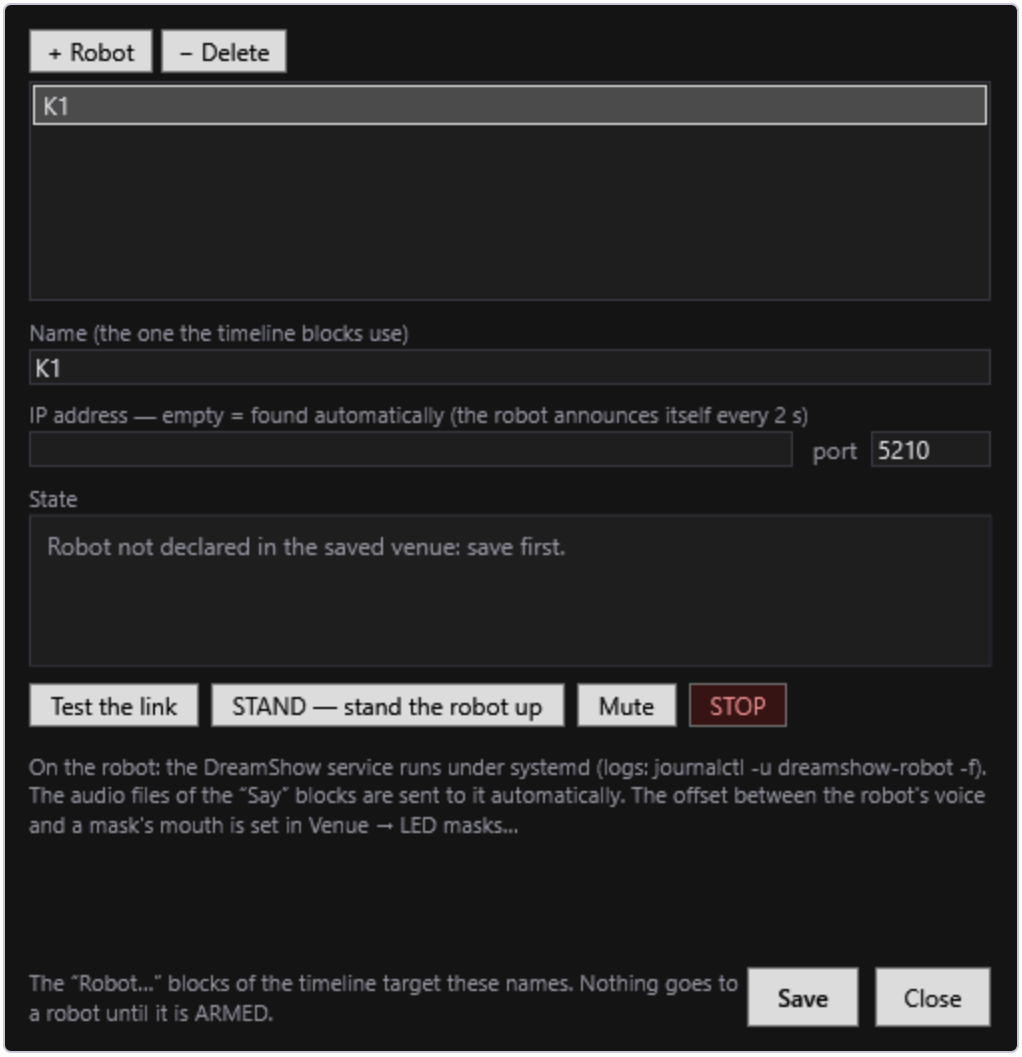
The service is installed once on the robot. It is supplied by Polux: ask support, who will tell you how to install it.

**To open it:** menu **Venue > Robots…** (tooltip: “Humanoid robots (Booster K1) driven by the DreamShow service installed on the robot. The “Robot…” blocks of the timeline drive them.”).

> ⚠ A humanoid robot is heavy and can fall. Always keep 2 m clear around it, a hard, flat floor, and the robot's controller in your hand during tests.

Several labels of the robot windows are not translated in the software and stay in French on screen; they are given below in French, with their translation in brackets the first time.

## The Robots window




The title reminds you of the venue: "Robots: <venue>". An asterisk * flags unsaved changes.

- The **list** of the robots declared in the venue.
- **+ Robot**: adds a robot named "K1" (then "K1 2"…) and selects it.
- **− Delete**: removes the selected robot.
- **Name (the one the timeline blocks use)**: the name the Robot… blocks use to target this robot.
- **IP address — empty = found automatically (the robot announces itself every 2 s)**: leave empty in the usual case. The robot's service announces itself on the network; DreamShow keeps the address of the announcement that carries the same name, or, if the venue has only one robot, that of any robot that announces itself. An address typed here turns off the automatic search.
- **port** (tooltip "TCP port of the service (5210 by default)"): only change it if the robot's service has been set up differently.

### State

The **State** frame updates continuously (for the selected robot):

- “Robot not declared in the saved venue: save first.”: click **Save**.
- “Waiting for the robot to announce itself on the network (UDP 5211)… Is the dreamshow_robot.py service running? Same network as the console?”: DreamShow has not yet received any announcement.
- “Unreachable: address:port” followed by the error, in red: the address is known but the robot does not answer.
- “Connected address:port”, in green, then:
    - “(SIMULATION, no SDK)” if the service runs without driving a real robot;
    - **Mode** (for example Damping for damped, Prepare for standing) and **Batterie** (Battery) in %;
    - “⚠ AU SOL” (ON THE FLOOR) in red if the robot has fallen;
    - “En cours : …” (in progress: the current action) or “Inactif” (idle);
    - “ARMED” or “disarmed”;
    - idling: “idling stopped”, “idle: head” or “idle: head + arms”, with “(waiting for run mode)” if it is waiting for walking mode;
    - “Last error: …” where applicable.

### The test buttons

- **Test the link**: sends a simple call to the robot (it goes through even when disarmed).
- **STAND — stand the robot up** (tooltip “Puts the robot in standing mode (PREP). Asks for confirmation.”): a warning window reminds you that the robot is going to STIFFEN and take the standing position, that you must put it on the ground with its feet flat or hold it by the handle on its back, clear 2 m, keep the controller in your hand (LT + BACK puts it back to damping), that it holds its posture but does not catch an imbalance (it is the walking mode, LT + RT + A, that does that), and that it must be ARMED for the order to go out. **OK** sends the order, **Cancel** (the default choice) sends nothing.
- **Mute**: stops the sound the robot is playing (goes through even when disarmed).
- **STOP** (in red): immediate stop of movement and sound, even when disarmed.

The text at the bottom reminds you that the service runs on the robot under systemd (logs: `journalctl -u dreamshow-robot -f`), that the audio files of the “Dire” (Say) blocks are sent automatically, and that the offset between the robot's voice and a mask's mouth is set in **Venue > LED masks…**.

- **Save**: saves the robots in the venue (robots with no name are removed) and restarts the link.
- **Close**: if changes remain, DreamShow asks “Changes not saved. Save?”.

At the bottom of the window: “The “Robot…” blocks of the timeline target these names. Nothing goes to a robot until it is ARMED.”

## Test mode

Before putting a move into a show, try it here to see whether it **shifts the robot**. A kick, a dance or being picked up can make it slide or turn, and the rest of the show then plays in the wrong place.

1. Put the robot on its mark, then click 📍 **Mark its position**. It remembers where it stands, without moving.
2. Pick a move in the list (kicks, dances, gestures, get up, get ready), then click ▶ **Try**. A window describes the move and recalls the precautions before the order goes out.
3. Read under the buttons how far it moved, for example "From its mark: 23 cm forward, 8 cm to the left, turned 15° to the left". Beyond 10 cm or 10°, the line turns orange.
4. ↩ **Back to its mark** makes it walk straight to its mark, without turning round, then take back the heading it had; it corrects in two or three small passes at most. One return at a time: the button stays greyed out on the way. It must be in walk mode, and the mark 1.5 m away at most; further, put it back by hand.

If a move shifts the robot, add two **Robot…** blocks to the timeline: **Mark its position** at the start of the show, and **Back to its mark** after that move.

The position comes from the robot's odometry: expect a few centimetres of error, a little more after a sudden move. The robot must be **armed** for the moves to go out; marking its position works even disarmed, since it does not move it.

Test mode needs a recent version of the robot's service; with an older service, the window says so: ask the support for its update.

## Arming 🤖

As soon as the venue declares a robot, a button appears at the top right of the main screen, next to **LIVE**:

- 🤖 **disarmed** (normal appearance): the timeline's Robot blocks are **ignored**. You rehearse the light and the sound without the robot moving. The log notes each order set aside: "… DISARMED — command ignored: …".
- 🤖 **ARMED** (red background): the Robot blocks are executed.

One click switches from one to the other. The button also shows the state of the first robot: "· not found", "· hors ligne" (offline), or the mode and the battery (for example "· Prepare 82 %"), "⚠ AU SOL", "(simu)".

The log notes "Robots ARMED — the Robot blocks are executed" or "Robots disarmed — the Robot blocks are ignored".

DreamShow always starts **disarmed**: you have to arm deliberately at each session.

### In the cue window (Show mode)

The cue window also shows:

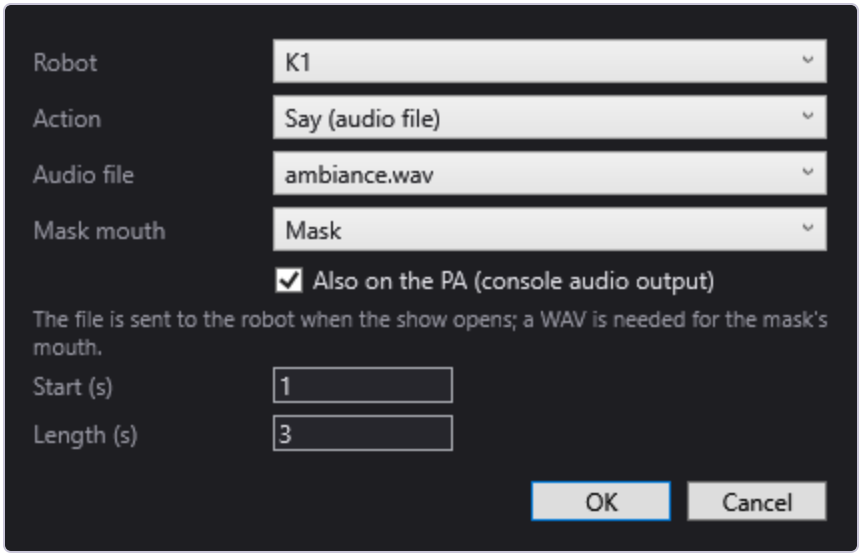
- the 🤖 **Robot disarmed** / 🤖 **Robot ARMED** button (red when armed), with the robot's state; a click arms or disarms;
- a red ⛔ **EMERGENCY — <name>** button per robot (and ⛔ **EMERGENCY — ALL** if there are several): “immediate stop then motors off (the robot slumps). Goes through even when disarmed.”.

> The emergency stop turns the motors off: the robot slumps. Only use it if the robot has to be neutralised immediately.

## The Robot… blocks

In the **Libraries** panel, the **Others** list contains **Robot…**. Drag it onto a timeline track: the **Robot block** window opens at once (if you cancel it, no block is created). To modify a block: double-click, or right-click > **Properties…**.

On the timeline, the block shows 🤖, the robot's name and the action, for example “🤖 K1 : Danse : Michael Jackson (partie 1)”, with “+ 🎭 <mask>” and “+ sono” for a line accompanied by sound.




### The common settings

- **Robot**: the robot targeted. If the venue declares none, the window says so.
- **Action** (action): the action (list below). Changing the action rebuilds the settings specific to the action.
- **Start (s)**: the moment, from the start of the scene, when the order goes out.
- **Length (s)**: the length of the block on the timeline. It is filled in automatically according to the action (and the length of the file for “Say”) as long as you do not change it yourself. It is used to place the following blocks; the robot, for its part, carries out its action at its own pace.
- **OK** / **Cancel**.

### The actions

| Action | Settings | Suggested length |
|---|---|---|
| **Prepare (ready to move)** | — | 5 s |
| **Se lever** (get up) | — | 8 s |
| **Lie down ⚠ not on K1** | — | 6 s |
| **Say (audio file)** | **Audio file**, **Mask mouth**, **Also on the PA (console audio output)** | length of the file |
| **Danser** (dance) | **Danse** (dance) | depends on the dance |
| **Boxing / kick** | **Mouvement** (movement) | 6 s |
| **Geste** (gesture) | **Geste** | depends on the gesture |
| **Head (tilt, rotation)** | **Tilt (°)**, **Rotation (°)** | 1 s |
| **Avancer / reculer (m)** (move forward / back) | **Distance (m)** | distance ÷ 0.2 m/s + 1 s |
| **Turn (°)** | **Angle (°)** | angle ÷ 25 °/s + 1 s |
| **Mode (debout, marche, amorti)** (mode: standing, walking, damping) | **Mode** | 1 to 3 s |
| **Veille (petits mouvements)** (idle: small movements) | **Veille** (idle) | 0 s |
| **Stop (movement and sound)** | — | 0 s |
| **Mute the robot** | — | 0 s |

Details of the actions:

- **Prepare (ready to move)**: puts the robot in the ready-to-move posture. It is essential before a dance, a kick, a gesture or a walk.
- **Se lever** (get up): raises the robot when seated or lying, on its back as well as on its front.
- **Lie down ⚠ not on K1**: the window warns in red that the K1 cannot lie down (the command is empty at the manufacturer's and a standing robot would risk falling): the service refuses it. Lay the robot down by hand, in damping mode.
- **Say (audio file)**:

    - **Audio file**: one of the sound files in the show's media folder (“No audio file in the show's media folder.” if there is none). The robot plays it through its own loudspeaker.
    - **Mask mouth**: “(none)” or an LED mask of the venue, whose mouth will move to that same file. The mouth requires a WAV file.
    - **Also on the PA (console audio output)**: also plays the file on the PC's sound output, at the same moment, like a sound block.
    - The reminder: “The file is sent to the robot when the show opens; a WAV is needed for the mask's mouth.”
- **Danser** (dance) — **Danse**: Michael Jackson (parts 1 and 2) 57.6 s, “Michael Jackson (partie 1)” 21.5 s, “Michael Jackson (partie 2)” 30.3 s, “Michael Jackson (partie 3)”, “Danse arabe” (Arabic dance), “Shan He Gu Ren”, “Gai Ge Chun Feng” (about 30 s each). “Robot standing and stable, 2 m clear. “~” = length to be measured.”: the lengths marked ~ are estimates.
- **Boxing / kick** — **Mouvement** (movement): Boxing kick, Roundhouse kick. “Robot standing and stable, 2 m clear: a kick moves the centre of gravity.”
- **Geste** (gesture) — **Geste**: Wave, Handshake (4 s), New year dance, “Danse Nezha”, “Danse Towards Future” (about 20 s), “Dab”, “Ultraman”, “Salut respectueux” (respectful bow), “Encouragement” (cheering), “Chat porte-bonheur” (lucky cat, 3 s).
- **Head (tilt, rotation)**: “Positive tilt = head down; positive rotation = to the left.”
- **Avancer / reculer (m)** (move forward / back): “Negative = backwards. Walks at 0.2 m/s, stops on odometry (± 5 cm).” Suggested value: 0.5 m.
- **Turn (°)**: “Positive = to the left.” Suggested value: 90°.
- **Mode (debout, marche, amorti)** (mode): “Debout (PREP)” (standing), “Marche (WALK)” (walking), **Damping (DAMP) — the robot slumps**.
- **Veille (petits mouvements)** (idle: small movements) — **Veille**: **Head only**, **Head and arms**, **Tête, bras et jambes** (head, arms and legs), **Stopped**. When idling, the robot slowly looks around when it has nothing else to do (the head turns up to 18°), to seem alive; with the arms, a 3.5° breathing at the shoulders and a slow 7° sway are added, faded in over 6 seconds; with the legs (this choice includes the arms), the robot drives no leg joint — that is impossible on the K1 — but every 15 to 35 seconds it briefly hands the arms back to its own controller and takes them again, which makes its balance settle again: the hips and the lower body really move, without it leaving its mark. Idling pauses during each action, requires walking mode, and stops on a STOP.
- **Stop (movement and sound)**: “Immediate stop of movement and sound (sent even when disarmed).”
- **Mute the robot**: stops the sound only.

## Sound files and masks' mouths

- When the show opens (and after each change), DreamShow sends the robot, in the background, the files of the “Dire” (Say) blocks it does not yet have, as soon as the robot is found on the network. The log notes ““…” sent to K1”.

- For a line accompanied by a mask's mouth, MaskCtl calculates the mouth sequence from the WAV file (see LED masks). If the file is not a WAV, the log flags it: “the mask's mouth needs a WAV file (MaskCtl does not read MP3)”.
- The **Robot voice → mouth offset (ms):**, set in **Venue > LED masks…**, aligns the mouth with the robot's real voice: positive, the mouth starts later; negative, the order to the robot goes out earlier.

## Step by step

### Making the robot wave and speak at curtain-up

1. Switch the robot on, check that it is on the same network as the PC.
2. Open **Venue > Robots…**, click **+ Robot**, then **Save**. Wait for “Connected …” in **State**; click **Test the link**.
3. Copy the line (preferably as a WAV) into the show's media folder.
4. In the scene, drag **Robot…** onto a track at 0 s: action **Se lever**, **OK**.
5. At 8 s, add a **Prepare (ready to move)** block.
6. At 13 s, a **Geste** > **Wave** block.
7. At 17 s, a **Say (audio file)** block with the line; tick **Also on the PA** if the audience must hear it through the sound system.
8. Rehearse disarmed: the light and the sound run, the robot does not move.
9. Robot placed on the ground, 2 m clear, controller in hand: click 🤖 **disarmed** to switch to 🤖 **ARMED**, and replay the scene.

## Good to know

- **Safety**: when disarmed, no movement order goes out; only the test call, **Mute** and **STOP** always go through. The emergency stop in the cue window also always goes through.
- An order sent to an unreachable robot is lost: “… unreachable — command lost: …” in the log.
- The lengths marked “~” are estimates: measure them on your robot and adjust the length of the blocks.
- The show can be prepared and simulated without a robot.
- At first start, the Windows firewall may ask you to allow DreamShow on the network: accept (ports used: TCP 5210 to the robot, UDP 5211 for its announcements).

## See also

- LED masks
- The scene timeline
- Show mode and the cue window
- Creating a show with an AI

- Troubleshooting

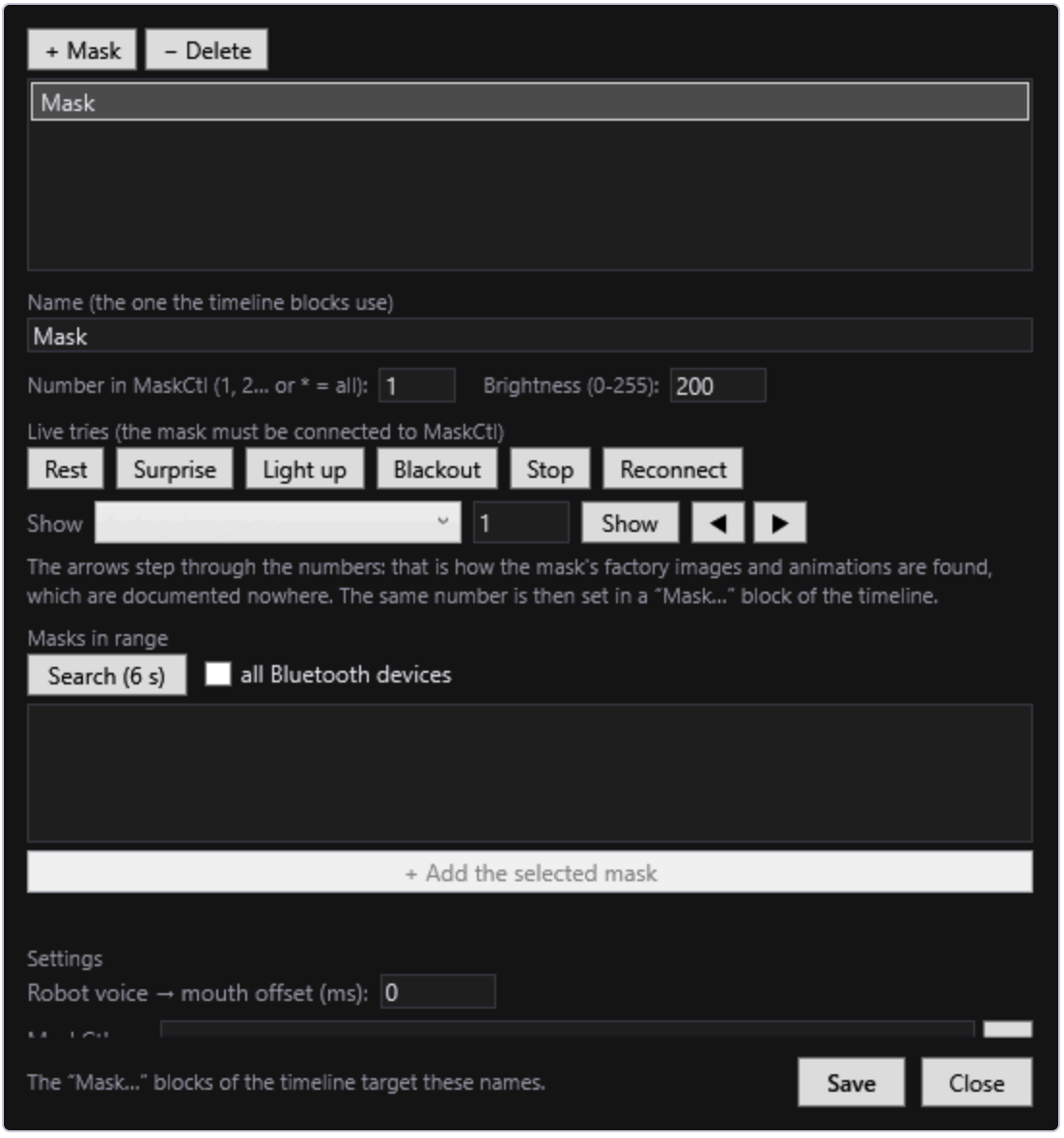
- Troubleshooting

# LED masks

DreamShow drives Bluetooth LED masks of the Shining Mask family, worn by an actor or placed on a robot's face. From the timeline, a **Mask…** block changes the face displayed, shows a picture or an animation, scrolls a text, or makes the mask **speak**: its mouth moves in rhythm with a sound file.

The masks are driven by **MaskCtl**, a small program installed with DreamShow (in the software's `tools` folder), which connects to the masks over Bluetooth. DreamShow starts it by itself when the venue declares masks.

**To open it:** menu **Venue > LED masks…** (tooltip: “LED masks (Shining Mask) driven over Bluetooth by MaskCtl. The “Mask…” blocks of the timeline drive them.”).



## Before you start

As the bottom of the window reminds you:

- the mask must be **switched on** and **not paired in Windows** (do not add it in Windows' Bluetooth settings);
- MaskCtl connects to it over Bluetooth, **one link at a time**: close the mask's mobile app;
- the catalogue faces take the 20 stored slots of the mask ("faces upload" in MaskCtl);
- a "Speak" block needs a **WAV** file: the mouth is computed by MaskCtl when the show opens.

## The LED masks window

The title recalls the venue: "LED masks: <venue>". An asterisk * flags unsaved changes.

### The list of masks

- **+ Mask**: adds a mask named "Mask" (then "Mask 2"…), with the next number in MaskCtl.
- **− Delete**: removes the selected mask.
- **Name (the one the timeline blocks use)**: the name that the Mask… blocks use.
- **Number in MaskCtl (1, 2… or = *all*):** the mask's number in MaskCtl's list (`masks.txt` file next to `MaskCtl.exe`); `*` targets all the masks at once.
- **Brightness (0-255):** the brightness used by the **Light up** test button (200 by default).

DreamShow passes the **Mask…** blocks' orders to MaskCtl by itself: there is nothing to declare other than the masks themselves.

### Live tests (the mask must be connected to MaskCtl)

Buttons to test the selected mask right away:

- **Rest**: shows the rest face.
- **Surprise**: shows the Surprise face.
- **Light up**: switches the mask back on at the set brightness.
- **Blackout**: switches the display off (brightness 0).
- **Stop**: "Stops a running sequence or speech".
- **Reconnect**: "MaskCtl: reconnects the missing masks".

The **Show** line is used to explore the mask's content:

- choose **factory image no.**, **factory animation no.** or **stored image no.**;
- type a number (0 to 255) and click **Show**;
- ◀ and ▶ go to the previous or next number and show it at once.

As the window explains: "The arrows step through the numbers: that is how the mask's factory images and animations are found, which are documented nowhere. The same number is then set in

a “Mask…” block of the timeline.”

### Masks in range

- **Search (6 s)**: starts MaskCtl's Bluetooth search (“Bluetooth search running…”). Tooltip: “A mask ALREADY connected does not appear: it has stopped announcing itself.”
- **all Bluetooth devices**: widens the search to all devices, because “A brand-new mask does not necessarily announce itself under a name starting with MASK.”
- The list shows each device found with the signal strength, for example `MASK-BF5F1A -54 dBm` (the closer the number is to 0, the stronger the signal).
- **+ Add the selected mask**: registers the mask in MaskCtl's list (`masks.txt`), which sets its number, and declares it here under its Bluetooth name. Message: ““…” registered as no. …. Save, then “Reconnect” to attach it.” If this number is already declared, the window says so and selects the existing mask.

If nothing is found: “No mask found. Is the mask switched on, and disconnected from the mobile app? A mask already driven by MaskCtl no longer announces itself: it is normal that it does not appear.”

### Settings

- **Robot voice → mouth offset (ms):** used when a robot says a line and a mask moves its mouth on the same file. Tooltip: “Measured once: positive if the robot's voice comes out AFTER the command (the mouth is delayed by as much), negative if the order to the robot must go out earlier.” Default value: 0.
- **MaskCtl.exe:** the path of the MaskCtl program, with the **…** button to choose it. Left empty, DreamShow uses the copy installed with it (`tools\MaskCtl.exe` in the software folder). A relative path starts from the data folder. Tooltip: “Path to MaskCtl.exe (Bluetooth driver for the masks). Started automatically when the venue declares masks.”

At the bottom: “The “Mask…” blocks of the timeline target these names.”

- **Save**: saves the masks and settings in the venue, then restarts MaskCtl if needed.
- **Close**: if changes remain, DreamShow asks “Changes not saved. Save?”.

## MaskCtl started automatically

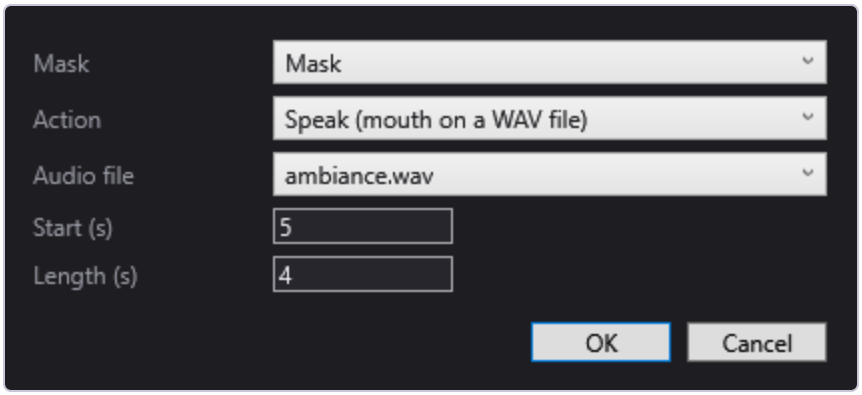
When the venue declares at least one mask and no MaskCtl is running, DreamShow starts it, minimised in the taskbar. The log notes “MaskCtl started automatically (pipe maskctl)”. Possible problems, flagged in the log:

- “MaskCtl.exe not found: …” or “MaskCtl.exe is not set (Venue → LED masks…): the masks will not be driven”;
- “Cannot start MaskCtl: …”.

## The Mask… blocks

In the **Libraries** panel, the **Others** list contains **Mask…**. Drag it onto a track: the **Mask block** window opens at once (if you cancel it, no block is created). To edit a block: double-click, or right-click > **Properties…**.

On the timeline, the block shows 🎭, the mask's name and the action, for example “ 🎭 Mask: Face: Anger”.




### The common settings

- **Mask**: the targeted mask.
- **Action**: the action (below).
- **Start (s)**: the moment, counted from the start of the scene, when the order is sent.
- **Length (s)**: the block's length on the timeline; for **Speak**, it follows the file's length as long as you do not change it.
- **OK** / **Cancel**.

### The actions

- **Catalogue face** — **Face**: one of the 20 faces of MaskCtl's catalogue, shown with their key:
  - the **mouths** (to compose speech by hand): Rest (rest), Closed lips (m/b/p), Clenched teeth (s/z), Lip and teeth (f/v), Tongue on teeth (th), High tongue (n/d/t), short i (ih), r (rr), small circle “oo” (oo), wide flat i/e (ee), open e (eh), oval “uh” (uh), l tongue raised (ll), medium round o (oh), open o (aw), wide a (ah), wide-open a (aa);
  - the **expressions**: Anger (colere), Surprise (surprise), Sadness (triste).
- **Stored image (number)** — **Stored image no.**: “0-19 = the faces of the MaskCtl catalogue; beyond that, your own stored images.”
- **Factory image (number)** — **Factory image no.**: a picture built into the mask.
- **Factory animation (number)** — **Factory animation no.**: an animation built into the mask. For these two actions: “Number in the mask's internal catalogue. Step through them in Venue → LED masks… (arrows ◀ ▶).”
- **Sequence of stored images** — **“Emplacements”** (Slots): “Numbers separated by commas (10 at most): the mask chains the images on its own, at the set speed.”

- **Speak (mouth on a WAV file)** — **Audio file**: a WAV file from the show's media folder. The list only offers WAV files; if there are none: “No WAV in the show's media folder (the mouth needs a WAV).”. The mouth moves in rhythm with this file; **the sound itself is not played by this block** (see below).
- **Animated face (blinking)**: a living face that blinks its eyes.
- **Stop the animation / the speech**: stops the running image sequence, animation or speech.
- **Scrolling text** — **Text**: a text that scrolls on the mask.
- **Brightness** — **Level (0-255)**: 200 by default.
- **Scrolling speed** — **Speed (0-255)**: speed of the text and of image sequences; 50 by default.
- **“Mode d'affichage”** (Display mode) — **Mode**: **“Fixe”** (Static), **“Clignotant”** (Blinking), **Scroll left**, **Scroll right**, **Custom static**.
- **Blackout (switch off)**: switches the display off (brightness 0).

## Lip sync

The mouth “speaks” from a sequence computed by MaskCtl on a **WAV** file:

1. When the show is opened, and after each change, DreamShow spots the **Speak** blocks and the robot lines that animate a mask's mouth.
2. For each WAV file that does not yet have its sequence, it asks MaskCtl to compute it. The log notes “Mouth analysis by MaskCtl: …”. MaskCtl writes a file of the same name with the `.seq` extension next to the WAV.
3. When the block passes, the mask plays this sequence.

Two ways of making a character speak:

- **A masked actor**: place the sound file in a sound block (for the PA) **and** a Mask **Speak** block with the same file, at the same start.
- **A robot with a mask**: a single Robot **Say (audio file)** block, with **Mask mouth** set to this mask (see Humanoid robots). The **Robot voice → mouth offset (ms)** then aligns the mouth with the robot's real voice; it only applies to this case.

### The voice / mouth offset

A robot's loudspeaker may output the sound a little after the order. To compensate:

1. Have the robot say a line with the mask's mouth, and watch.
2. If the mouth is **ahead of** the voice, set a **positive** offset (in milliseconds): the mouth starts that much later.
3. If the mouth is **behind**, set a **negative** offset: the order goes out to the robot sooner.
4. **Save** and replay until the two coincide.

## Step by step

### Declaring a brand-new mask

1. Switch the mask on, close its mobile app.
2. Open **Venue > LED masks...**, click **Search (6 s)** (tick **all Bluetooth devices** if nothing appears).
3. Select the mask found, click **+ Add the selected mask**, then **Save**.
4. Click **Reconnect**: MaskCtl attaches to the mask.
5. Test with **Rest**, **Surprise**, **Blackout**, **Light up**.

### Making a masked actor speak

1. Convert the line to WAV and copy it into the show's media folder.
2. Place the WAV on a sound track at the moment you want.
3. On another track, at the same start, drag **Mask...**, action **Speak (mouth on a WAV file)**, choose the same file, **OK**.
4. Wait a few seconds for the log to report the mouth analysis, then play the scene.

## Good to know

- MaskCtl does not read MP3 files: the mouth needs a WAV. Creating a show with an AI makes this WAV by itself from an m4a or an mp3.
- If you replace a WAV with a new version of the same name, delete its `.seq` file: DreamShow only recomputes the sequence if it does not exist.
- A mask already connected to MaskCtl does not appear in the search: this is normal.
- The show can be prepared and simulated without a mask.

## See also

- Humanoid robots
- The scene timeline
- Sound and video on the timeline
- Creating a show with an AI
- Troubleshooting

# Connected sockets

DreamShow can switch power sockets on or off during a show: cut the power of architectural fixtures at night, bring it back before opening, start a smoke machine or a sign at the right moment. Four kinds of sockets are supported:

| Type | What you need | Socket setting |
|---|---|---|
| **Home Assistant** | a Home Assistant server that already knows the socket (Zigbee through ZHA or Zigbee2MQTT, Wi-Fi…) | the entity, for example `switch.facade_lights` |
| **Zigbee2MQTT** | a Zigbee USB stick, Zigbee2MQTT and an MQTT server (Mosquitto…), without Home Assistant | the device topic, for example `zigbee2mqtt/facade_plug` |
| **Shelly** (Wi-Fi) | nothing else: DreamShow talks to the socket directly | its IP address |
| **Tasmota** (Wi-Fi) | nothing else: DreamShow talks to the socket directly | its IP address |

**To open it:** menu **Venue > Connected sockets…**.

## Gateways

At the top of the window, the settings shared by all sockets:

- **Home Assistant: address** (for example `http://homeassistant.local:8123`) and **Access token**. Create the token in Home Assistant: your profile → Security → Long-lived access tokens → Create token. It is saved in the venue file.
- **MQTT server: address**, **port** (usually 1883), **MQTT user name** and **MQTT password**: the MQTT server used by Zigbee2MQTT. Leave the user name empty if it does not ask for one.

Shelly and Tasmota sockets need none of this.

## Sockets

- **+ Socket** adds a socket, **− Delete** removes the selected one.
- **Name**: the name chosen in timeline blocks (two sockets cannot share a name).
- **Type**: Home Assistant, Zigbee2MQTT (MQTT), Shelly (Wi-Fi) or Tasmota (Wi-Fi).

- **Entity**, **Zigbee2MQTT topic** or **IP address**, depending on the type. Make the IP address of Wi-Fi sockets fixed in your router, otherwise it may change.
- **Relay no.** (Shelly and Tasmota): 0 for the first relay; useful for multi-relay modules, for example a module in the switchboard that cuts two lines of fixtures.
- **On**, **Off**, **Read state**: immediate test of the selected socket with the current settings (even unsaved). The result is shown next to the buttons: « Socket: on », « Socket: off » or the reason for a failure (token refused, no answer, address not set…).
- **Save** stores the list in the venue.

For Shelly, DreamShow recognises both generations of devices by itself (the « RPC » API of Shelly Plus, Pro and Gen3, and the API of the first Shelly devices).

## The Socket… block

In **Libraries**, **Others** list, drag **Socket…** onto a timeline track. Choose the socket and the action: **on**, **off** or **toggle** (inverts the current state). On the timeline, the block shows 🔌, the socket name and the action.

The command is sent when the playhead reaches the block, in show mode as well as during playback in the editor. It never holds up the show: if the socket does not answer, the log says so (category « sockets ») and the show goes on.

## In the conductor window

While the conductor window is open, a 🔌 **Sockets** strip, at the bottom of the large left panel, lists the venue's sockets with their state, read again every 10 seconds: ● **on** (green), ○ **off**, ⚠ **unreachable** (orange: the socket or its gateway does not answer) or ◌ **state unknown** (not read yet). Zigbee2MQTT sockets also report each change as it happens.

## Architectural lighting: an example

For a façade that lights up at sunset and goes off at 1 am:

1. A **Wait for a time, a date, the sun…** block set to sunset, then a **Socket…** block « Façade » set to **on**, then, one or two seconds later, the light blocks (time for the fixtures to start).
2. Further on, a **Wait…** block set to 1:00, light blocks at zero, then a **Socket…** block « Façade » set to **off**.

## Good to know

- The control PC must be on the same network as the Wi-Fi sockets, the Home Assistant server or the MQTT server.

- With Zigbee2MQTT, the socket's real state comes back over MQTT: the window shows it after **Read state**.
- The Home Assistant token and the MQTT password are stored in plain text in the venue file: do not share this file outside your team.

## See also

- Waits
- Architectural lighting

# Shows, files and projects

This page describes the **File** menu (creating, opening, saving a show, exporting or importing a complete project) and explains where DreamShow stores your venues, your shows, your media and your settings.

**To open it: File** menu. Shortcut: Ctrl+S to save.

## Show, venue, project: what are we talking about?

- A **show** is your performance: the sequence of scenes, their timelines and their final states. It is a `<name>.show.json` file.
- A **venue** describes the place and the equipment: fixtures, addresses, positions, groups, screens, DMX connections. It is a `<name>.venue.json` file. A venue can serve several shows.
- A show is linked to its venue **by the venue's name**. When you open a show, DreamShow automatically switches to its venue (if it exists).
- The **project** is the whole DreamShow data folder: all the venues, all the shows, the media, the effects, the palettes, the trajectories and the settings.

## The File menu

### New show…

Creates an empty show on the active venue.

1. If the current show has unsaved changes, DreamShow asks **Save the changes to the show?**.
2. A small **New show** window asks for **Show name:**. Type the name then **OK**. Closing this window without a name cancels.
3. If a show of that name already exists: “A show “…” already exists. Overwrite it?”. **No** cancels.
4. The show is created with a single scene, “Scene 1”, saved straight away and opened.

The file is created in the same folder as the current show.

### Open a show…

Opens a `*.show.json` file (file picker, filter “Show DreamShow” (DreamShow show)), starting from the current show's folder. DreamShow first offers to save the current show if it has been modified.

On opening:

- if the show was prepared for a venue other than the active one, DreamShow switches to that venue; if it cannot be found, it keeps the current venue and writes this in the **Log** (“The show references venue “…”, not found — current venue kept (…)”);
- the first scene is selected and playback is reset to zero;
- the show is added at the top of the **Recent shows**.

### Recent shows

The last 10 shows opened, most recent first, numbered (“1 name.show.json”, “2 …”): type the number to open it from the keyboard while the submenu is open. The open show is in bold. The tooltip gives the full path. **(none)** is displayed when the list is empty.

A file that no longer exists is removed from the list with the message “File not found: …”.

### Save (Ctrl+S)

Saves the current show. A message “✓ show saved (HH:mm:ss)” is displayed for a few seconds next to the show's title, and the asterisk `*` that flagged changes disappears.

At each save, DreamShow:

- first writes a temporary file, then replaces the old one: a power cut during the save cannot destroy the previous version;
- keeps a copy of the previous version in the `autosave` subfolder (see below);
- notes in the show the name of the active venue (this is the one that will be reopened with this show).

### Save as…

Saves the show under another name (filter “Show DreamShow”, extension `.show.json` added if needed), then carries on working on this new file.

Each show has its own media folder: DreamShow **copies** the old show's media into the media folder of the new name (only the files that are not already there). A message says “… media copied to …”. The show saved under another name is thus complete and independent of the original.

### Remote (GO keys)…

Assign the keys of a 2.4 GHz remote to the cue window's GO / Previous step / Blackout actions. See The remote (GO keys).

### Open the show's media folder

Opens in Windows Explorer the media folder of the current show (`data\medias\<show name>`), creating it if it does not exist. This is where the sounds, videos and DMX streams used by the show

go. Media are called **by name only**, never by a full path: a project moves from one PC to another without breaking anything.

To add media, preferably use the **Import media…** button of the media panel, which copies them into this folder. See Sound and video on the timeline.

### Export the whole project (zip)…

Venue patch, fixtures, shows, effects and media — to transfer to another PC.

1. If the current show has changes, it is saved first, without asking.
2. Choose where to save the zip file (suggested name: `<venue name>-dreamshow.zip`, spaces replaced by hyphens).
3. DreamShow puts in the zip **the whole data folder**, subfolders included, except the copies in the `autosave` folder and the screen layout (`dreamshow.ui.json`).
4. A message gives the zip's size and location: “Project exported (… MB): …”.

### Import a project (zip)…

Loads a project exported from another PC (or a backup).

1. Choose the zip file (filter “Projet DreamShow” (DreamShow project)).
2. Confirm: “The import replaces the files of the current project that have the same name (venue, shows, effects, media). Continue?”
3. The zip's files are copied into the data folder, replacing those with the same name. Files that are not in the zip stay in place.
4. “Project imported — the application will restart to load it.”: DreamShow restarts by itself.

### Create a show with an AI…

See Creating a show with an AI.

### DMX stream by an AI…

Request to paste into an AI chat (venue, fixtures, music), then checked import of the stream it returns. See DMX stream by an AI.

### Export for a standalone player…

DMX stream (.DmxStreaming) replayed without a PC by an ESP32, a Raspberry Pi or a mini-PC. See The standalone player.

### Import a ppshow show…

For users of the ppshow software: rebuilds a DreamShow show from a ppshow show folder (the `dmx.ini` file and the sounds), matching the fixtures **by DMX address** to the current venue. The active venue must therefore reflect the ppshow rig exactly.

1. Choose the ppshow show folder (picker title: “ppshow show folder (containing dmx.ini)”). Without `dmx.ini`: “This folder has no dmx.ini (it is not a ppshow show).”
2. Give the **DreamShow show name:** (the folder name is suggested). If that show already exists, DreamShow asks whether to overwrite it.
3. The sounds are copied into the new show's media folder.
4. A summary is displayed: number of ppshow steps and scenes created (steps in automatic sequence may be merged), fixture and moving-head states matched, channel states (dimmers), named steps, extra commands, sounds and sound level changes. ppshow fixtures with no DreamShow fixture at the same address are ignored and counted.

## Where files are stored

Everything is in the `data` folder, next to `DreamShow.exe` (in a normal installation: `C:\Users\<you>\AppData\Local\Programs\DreamShow\data`). The simplest way to get there: **File > Open the show's media folder**, then go up two levels.

| Item | Location |
|---|---|
| Venues | `data\<name>.venue.json` (one per venue) |
| Shows | `data\<name>.show.json` |
| A show's media | `data\medias\<show name>\` (one folder per show) |
| Effects | `data\effects.json` (shared by all shows) |
| Palettes | `data\palettes.json` (shared by all shows) |
| Trajectories | `data\trajectoires\<name>.traj.json` (one per trajectory) |
| Fixture definitions | `data\fixtures\`; your edited definitions in `data\fixtures\custom\`, those updated online in `data\fixtures\online\` |
| Copies of previous versions | `data\autosave\` |
| DreamShow settings | `data\dreamshow.settings.json` (language, recent shows, GO keys, MIDI, OSC, web remote…) |
| Screen layout | `data\dreamshow.ui.json` |
| Translations | `data\lang\` |

| | |
|---|---|
| Crash-recovery marker | `recovery.json`, in the `DreamShow` folder of your Windows account's local data (`%LOCALAPPDATA%\DreamShow`) (see Recovering after a crash or power cut) |

### The autosave folder: previous versions

Each time a show, a venue or the palettes are saved, DreamShow first copies the version that is about to be replaced into `data\autosave`, with the date and time in the name. For example, saving `concert.show.json` on 5 October 2026 at 14:32:10 keeps the old version as `concert.show-20261005-143210.json`.

DreamShow does not save the show by itself at regular intervals: remember Ctrl+S. The `autosave` folder is never emptied automatically.

## Step by step

### Starting from an existing show for a new performance

1. Open the starting show (**File > Open a show...**).
2. **File > Save as...**, type the new name.
3. You are now working on the copy, with its own copy of the media. The original is not modified.

### Transferring a project to the show control PC

1. On the preparation PC: **File > Export the whole project (zip)...**, save the zip on a USB stick.
2. On the show control PC (DreamShow installed): **File > Import a project (zip)...**, choose the zip, confirm.
3. DreamShow restarts. Open the show with **File > Open a show...** if it is not the one that opens.
4. Check **Venue > DMX connections...**: the show control PC's interfaces may be different.

### Going back to a previous version of a show

1. In DreamShow, open another show (or quit DreamShow), so as not to work on the file you are about to replace.
2. In Explorer, open the `data` folder. To keep the current version, rename for example `concert.show.json` to `concert-actuel.show.json`.
3. Open the `autosave` subfolder and find the copy with the right date and time, for example `concert.show-20261005-143210.json`.
4. Copy it into the `data` folder and rename it exactly like the original show: `concert.show.json`. It thus finds its media folder `medias\concert` again.
5. In DreamShow, **File > Open a show...** and choose `concert.show.json`.

## Good to know

- The asterisk `*` after the show's title flags unsaved changes. DreamShow asks to save before opening another show, creating one, launching an assistant and when quitting.
- When quitting, DreamShow asks “Quit DreamShow?” (or “A show is RUNNING. Quit DreamShow anyway?” if the cue window is open).
- A show's media folder depends on **its file name**. If you rename a `.show.json` file in Explorer, also rename its folder in `medias`, or rather use **Save as…**.
- A show is linked to its venue by the venue's name. Rename a venue with **Venue > Rename the venue…**, which updates the shows that reference it: see Switching, renaming, duplicating a venue.
- The project export contains the whole data folder, including the fixture library and your settings (language, keys, MIDI, OSC…). The import therefore also replaces the settings of the PC that receives it.
- Save your show before importing a project: at the restart that follows the import, DreamShow offers to save the current show, and answering **Yes** would overwrite this show on top of a file of the same name that the import has just brought in.
- Before updating DreamShow, export the project: see Installation, languages and updates.

## See also

- Recovering after a crash or power cut
- Switching, renaming, duplicating a venue
- Sound and video on the timeline
- Libraries
- Installation, languages and updates

# Troubleshooting

This page gathers the most frequent problems and the steps to follow for each. The first reflex is always the same: look at the **Log** panel, where DreamShow notes what it is doing and what is going wrong (**AVT** lines for warnings, **ERR** for errors).

## The log, your first tool

The **Log** panel shows DreamShow's messages, with the time and the source (dmx, video, audio, robot, masks, displays…).

- The **Info**, **Warn.**, **Errors** and **Debug** boxes choose which messages are displayed (Debug, very chatty, is unticked by default).
- **Clear** empties the display.
- **Copy**: "Copies the displayed journal to the clipboard". Paste these lines into a request for help: they say exactly what happened.
- **Follow** scrolls the log to the most recent messages.

See Log, Audio and Clock.

## Nothing goes out to the fixtures (DMX)

Check in this order:

1. **The LIVE button**, at the top right. When on, the selected scene is sent to the interfaces; when off, "no frame goes out any more, only the simulation keeps running". It has no effect in show mode.
2. **Visualiser mode.** During **DMX > Visualiser mode (Art-Net / sACN in)…**, DreamShow stays silent: its output is cut. Stop this mode.
3. **The connections.** Next to the **Connections…** button (at the top of the screen), a text summarises the outputs in use; its tooltip lists any errors. Open **Venue > DMX connections…** and check that each universe in use has an interface (and not "None"). See DMX connections.
4. **What DreamShow sends.** Open **DMX > DMX values…**: if the values move when the scene plays, DreamShow is indeed sending; the problem is between the PC and the fixtures.
5. **Depending on the interface:**
    - **Art-Net**: the node's IP address must be on the same network as the PC, and the network universe number must match the one set in the node. The Windows firewall must allow DreamShow (see below).

    - **sACN**: check the universe number (1 to 63,999), and that the network lets multicast through, or give the receiver's IP address.
    - **USB DMX PRO**: the interface's COM port must appear in the ""Ports COM : …" (COM ports: …)" list of the **DMX connections** window; otherwise, check the connection and the interface's driver.
    - **uDMX**: the interface uses the libusbK driver.
6. **The fixtures.** The DMX address and the mode (number of channels) set on each fixture must be those of **Venue > Fixtures, addresses, positions…**. Check the cables and the termination at the end of the line.
7. **Testing a channel. DMX > DMX faders…** forces channels by hand on top of the cue list: raise a fixture's intensity channel to check that it responds. See DMX values and DMX faders.




## A fixture is behaving erratically

- Wrong DMX mode: the fixture is set to a different number of channels from the one declared in the venue. Align the two.
- Overlapping addresses: two fixtures share channels. Check the addresses in **Venue > Fixtures, addresses, positions…**.
- Incorrect definition: correct it in **Venue > Fixture library…**. See The fixture library.
- A moving head aims badly: check its position, orientation and rigging in the venue, or calibrate it. See Moving-head calibration.

## The video does not start

Videos on monitors are played by **ppvideoplayer**, installed with DreamShow. DreamShow starts it by itself, **minimised in the taskbar**, when the show contains videos. Read the log (source "video"):

- "ppvideoplayer started automatically (the show contains videos)" then "ppvideoplayer ready — media folder and loads sent": all is well on the launch side.
- "The show has videos but ppvideoplayer cannot be found (…)": the player is not at the indicated locations; reinstall DreamShow.
- "ppvideoplayer started but its pipe does not answer (15 s) — new attempt in 30 s.": the player is starting badly; wait for the new attempt, or restart it (see below).
- "ppvideoplayer is running (…) but its pipe is missing — dead instance, it is stopped and restarted.": DreamShow repaired this by itself.

Also check:

1. That the video file is in the show's media folder (**File > Open the show's media folder**).

2. That the chosen screen exists: **Venue > Screens...**, 🔎 **Identify the screen** button (a test pattern appears on the monitor; a click or a key closes it), then ▶ **Test the video**.
3. **ppvideoplayer only sees the monitors that were connected when it started.** If you connect a screen afterwards, click ⟳ **Restart ppvideoplayer** in **Venue > Screens...**.

See Video screens and Sound and video on the timeline.

## No sound

- DreamShow plays the sounds itself, on the **Windows default audio output**: it has no output setting of its own. Change the default output in Windows' sound settings if needed.
- Look at the levels of the **Audio** panel (left / right): a **Sound level** block on the timeline may have lowered them.
- The log (source "audio") flags a missing file: "File not found: ...". The file must be in the show's media folder.
- 🔇 **Mute the sounds** (**Audio** panel, and cue window) immediately stops all sounds in progress.

See Log, Audio and Clock.

## DreamShow is slow, the simulation stutters

- **Turn the simulation off: the ⏻ button**, at the top of the **Venue** panel (the simulation). When off, "the room is no longer computed or drawn (2D and 3D), which relieves the processor and the graphics card during a show. The DMX output is not affected." The panel shows "Simulation off — the ⏻ button turns it back on. The DMX output carries on normally." This setting is kept at the next launch: if the simulation stays black, check this button.
- **Lower the 3D quality**: in 3D view, the list next to **2D / 3D** offers **Basic** (flat cones), **Standard** (gradient core and halo, soft impact spots) and **Maximal** (rendered by the graphics card, volumetric beams). Choose Basic on a modest PC.
- **Switch to 2D**, which is much lighter than 3D.
- The small line under the simulation shows the refresh rate and the cost of one frame; the bar at the top of the main screen shows the engine's real rate.
- The engine keeps playing the show even if the display is slow: the light is not slowed down by the simulation.

See 2D and 3D simulation.

## Windows firewall and network

Several DreamShow functions go over the network: Art-Net, sACN, the web remote, the OSC remote, follow-spot tablets, displays, IP video players, robots, visualiser mode. The first time, Windows may

ask whether to allow DreamShow on the network: **accept** (at least for private networks). If you refused, allow DreamShow in the Windows firewall settings.

Ports used (for a network administrator):

| Function | Port |
|---|---|
| Art-Net (output and visualiser mode) | UDP 6454 |
| sACN (output and visualiser mode) | UDP 5568 |
| OSC remote | 8000 by default (adjustable) |
| Web remote | 8092 by default (adjustable) |
| Follow-spot tablets | TCP 8090 |
| Backstage displays | TCP 8091 |
| IP video players | TCP 5555 |
| Robots | TCP 5210 (to the robot), UDP 5211 (announcements) |

For tablets and displays: the address shown by DreamShow is the PC's first network address. If it has several cards (Wi-Fi and cable, virtual card…), use its address on the devices' network.

## Tablets, displays, robots, masks

- **A follow-spot tablet does not connect**: same network, firewall, address with `:8090`. See Follow spot on a tablet.
- **A display stays empty**: check the zone name in the address, and that a Backstage display… block really targets this display. See Backstage displays.
- **The robot does not move**: is it **ARMED** ( button)? The log notes "… DISARMED — command ignored: …" otherwise. See Humanoid robots.
- **The mask does not respond**: switched on, not paired in Windows, mobile app closed, **Reconnect** button. See LED masks.

## After a crash or a power cut

Restart DreamShow. If it stopped during a show or an editing session (within the last 6 hours), it shows the **Resume after an interruption** window: "DreamShow stopped … (last point: …)." with the show, the scene and the position, and offers to **resume exactly at that point** (**Yes, resume** / **No**). See Recovering after a crash or power cut.

Your files are protected: each save also keeps a dated copy in an `autosave` subfolder of the data folder.

If the crash happens again, note the version (**? > About DreamShow...**), what you were doing, and copy the log (**Copy** button) for support.

## The screen layout is lost

The ↺ **UI** button, at the top right, restores the default layout of the panels. See Main screen overview.

## See also

- DMX connections
- DMX values and DMX faders
- Video screens
- 2D and 3D simulation
- Recovering after a crash or power cut
- Installation, languages and updates

# Keyboard shortcuts

This page lists all of DreamShow's keyboard and mouse shortcuts, window by window. The keys of the cue window (GO, previous, blackout) can be changed in **File > Remote (GO keys)…**; the others are fixed.

## Everywhere in the main screen (editor)

| Key | Action |
|---|---|
| **Space** | **GO**: moves on to the next scene, like the **GO (Space)** button of the **Scenes** panel. No effect when the cursor is in a text field, or on a check box or a drop-down list. |
| **Ctrl+S** | Saves the show (like **File > Save**). |
| *Key expected by a "Wait for a key…" block* | When the timeline is stopped on that block, the chosen key restarts playback. It takes priority over any other shortcut. |

In the editor, only the space bar does GO; the Enter key and the remote keys are only used in Show mode.

## Cue window (Show mode)

| Key | Action |
|---|---|
| **Space** or **Enter** (default) | **GO**. |
| **Backspace** (default) | ◀ **Previous**: goes back to the previous scene. |
| **N** (default) | **BLACKOUT**: general blackout, a second press releases it. |
| **F9** | Switches between the full window and the ▭ **Compact** one (small strip always on top). |
| **F11** | Full screen (borderless) or back to the normal window; leaves the compact window first if needed. |
| *Key expected by a "Wait for a key…" block* | Restarts the cue list; it takes priority over any other shortcut. |

The cue window's buttons remind you of the first key of each action: “GO (Space)”, “◀ Previous (Backspace)”, “BLACKOUT (N)”.

GO has no effect on the last scene and, in Manual GO, as long as the timeline of the current scene is not finished (the button is then greyed out).

### Changing the keys (remote)

**File > Remote (GO keys)…** opens the **Remote** window, with three sections: **GO (pas suivant)** (GO (next step)), **Previous step** and **Blackout**.

- **+ Key…** then press the key (or the button of a 2.4 GHz presentation remote, which sends keyboard keys): it is added to the action.
- **Esc** during the capture cancels it instead of adding the key.
- The small button on each key removes it.
- **OK** saves. GO always keeps at least one key: if you remove them all, Space is put back.

See The remote (GO keys).

## Scene timeline

| Gesture | Action |
|---|---|
| **Ctrl + wheel** | Horizontal zoom of the timeline, centred on the mouse. |
| **Wheel on the ruler** (at the top of the timeline) | Zoom, without Ctrl. |
| **Click on the ruler** | Places the red line at that moment. |
| **Drag a block** | Moves its start (0.1 s steps, snapped to the start or end of neighbouring blocks). |
| **Drag the right edge of a block** | Changes its length (0.1 s steps). |
| **Double-click on a block** | Opens its properties; for an effect block, switches to editing the effect; for a trajectory, opens its editor; for a dynamic effect or a pixel mapping, opens its settings window. |
| **Right-click on a block** | Menu: **Properties…**, **Delete**, and depending on the block **Edit the effect…**, **Edit the trajectory…**, **Replace with the current selection + settings**. |
| **Drag from Libraries** | A group, a media item, an effect, a trajectory or an item of the **Others** list dropped on a track creates a block. |

## Libraries

| Key or gesture | Action |
|---|---|
| **Del** in the **Trajectories** list | Deletes the selected trajectory (after confirmation). |
| **Double-click** on a trajectory | Opens its editor. |
| **Double-click** on an effect | Edits the effect. |
| **Right-click** on a media item | Media menu; the right-click first selects the item under the mouse. |

## Simulation (Venue panel)

**2D view (plan):**

| Gesture | Action |
|---|---|
| **Click on a fixture** | Selects or deselects it. |
| **Drag on the plan** | Rectangle selection: replaces the selection. |
| **Ctrl + drag** | Adds the fixtures in the rectangle to the selection. |
| With **Aim** pressed: **click-drag** | The selected moving heads follow the mouse. |
| With the **Zone** tool: **click-drag** | Draws, moves or resizes the zone. |

**3D view:**

| Gesture | Action |
|---|---|
| **Drag (left or right button)** | Orbits around the venue. |
| **Shift + drag**, or **drag with the wheel pressed** | Pans the view. |
| **Wheel** | Zoom in / out. |
| **Click (without moving) on a fixture** | Selects or deselects it. |
| With **Aim** or the **Zone** tool: **click-drag** | As in 2D, on the surface under the mouse. |

**Zone rotation**: in the rotation box (in degrees), **Enter** confirms the value.

See 2D and 3D simulation and Aiming and covering a zone with moving heads.

## Trajectory editor

| Key or gesture | Action |
|---|---|

| | |
|---|---|
| **Double-click** on an empty area of the keyframe strip | Adds a keyframe at that moment. |
| **Click** on a keyframe | Selects it; **dragging** moves it in time. |
| **Drag** on an empty area | Scrolls the preview. |
| **Enter** in the "Key — time (s):" box | Applies the time entered. |

See Trajectories.

## Fixtures, addresses, positions (venue editor)

| Key | Action |
|---|---|
| **Ctrl+Z** | Undo (↶ button). |
| **Ctrl+Y** | Redo (↷ button). |

In the **Add a fixture — QLC+ library** window: type your search, then **down arrow / up arrow** to browse the list without leaving the box, **Enter** (**Add** button) or **double-click** to choose.

## DMX values and DMX faders

| Key or gesture | Window | Action |
|---|---|---|
| **Double-click on a channel** | **DMX values…** | Opens the faders on that channel. |
| **Enter** in "First channel" | **DMX faders…** | Shows the page that starts at that channel. |
| **Enter** in a fader's value | **DMX faders…** | Applies the value typed. |
| **Wheel** on a fader | **DMX faders…** | ±1. |
| **Shift + wheel** | **DMX faders…** | ±10. |

See DMX values and DMX faders.

## Other windows

| Key | Window | Action |
|---|---|---|
| **T** | **Sound-to-light and tempo…** | Tap tempo (like the **TAP** button), except when the cursor is in a text field. |
| **Esc**, any key or a click | 🔎 **Identify the screen** test pattern (**Venue > Screens…**) | Closes the test pattern (it also closes by itself after a few seconds). |

| | | |
|---|---|---|
| *The key to wait for* | Properties of a **Wait for a key** block | After a click on **Set…**, the key pressed becomes the expected key. |

## In all dialog boxes

- **Enter** confirms the default button (**OK**, **Add**, **Save**… depending on the window).
- **Esc** closes without confirming (**Cancel** or **Close** button), in windows that have one.
- The usual text fields accept Ctrl+C, Ctrl+V, Ctrl+A, etc.

## See also

- Show mode and the cue window
- The remote (GO keys)
- The scene timeline
- Waits (key, time, date, sun)
- 2D and 3D simulation

# Glossary

The terms of the trade and those of DreamShow, with a short definition. Terms are in alphabetical order.

## A

- **Aim**: tool that points the selected moving heads at the point clicked on the plan or in the 3D view. See Aiming and covering a zone with moving heads.
- **Arm (a robot)**: allow the Robot blocks to make the robot move. When disarmed, they are ignored. See Humanoid robots.
- **Art-Net**: protocol that carries DMX over a computer network (Ethernet, Wi-Fi), to an Art-Net **node** that outputs it again on a DMX cable. Each universe has a network number (0 to 32,767 in DreamShow).
- **Audience in**: pre-show lighting mood: the house gently lit, the stage in half-light.
- **Auto**: trigger of a scene that moves on to the next one by itself after a delay.
- **Auto-fade (disengage)**: box that prevents any automatic chaining: everything advances on GO.
- **Autosave**: subfolder where DreamShow keeps a dated copy of each saved file.

## B

- **Backlight**: fixture placed behind the actors, lighting them from behind.
- **Backstage display**: information screen in the wings or dressing rooms, showing a web page served by DreamShow; the "Backstage display…" blocks of the timeline send messages to it. See Backstage displays.
- **Blackout**: the light switched off. **BLACKOUT** (cue window) is the general emergency blackout, which a second press releases; manual devices stay as they are.
- **Block**: item placed on a timeline track, with a start and often a length: light, effect, sound, video, wait, jump, robot, mask, display…
- **BPM**: beats per minute, the tempo of a piece of music.

## C

- **Calibration (of a moving head)**: setting that records where a moving head points at four corners of the stage (plus two optional points on the back wall), so that it aims accurately. See Moving-head calibration.

- **Cell**: part of a fixture that is set separately (each LED of a bar, each pixel of a matrix, each head of a multi-beam fixture).
- **Chained**: trigger of a scene that moves on to the next one as soon as everything it contains has finished.
- **Channel (DMX)**: one of the 512 values of a universe, from 0 to 255. A fixture uses one or several (intensity, red, green, blue, pan, tilt…).
- **Cue list**: the sequence of the show's scenes, in the order the stage manager sends them. By extension, the **cue window** of show mode.
- **Cue message**: text displayed large during a scene in show mode: what the stage manager is waiting for before pressing GO.

## D

- **DMX (DMX512)**: the standard language of lighting fixtures: a series of 512 channels, sent continuously, about 40 times a second in DreamShow.
- **DMX address**: number of the first channel a fixture uses in its universe (from 1 to 512). An 8-channel fixture at address 17 occupies channels 17 to 24. The address set on the fixture must be the one declared in the venue.
- **DMX faders**: window that forces channels by hand on top of the cue list. See DMX values and DMX faders.
- **DMX mode**: fixture configuration that sets its number of channels and their role (for example 8 channels or 16 channels). The mode chosen in the venue must be the one set on the device.
- **DMX stream**: DMX recorded frame by frame (.DmxStreaming file), played by a ▦ block of the timeline. See DMX stream by an AI.
- **DMX values**: window that shows live the 512 channels of each universe as they are sent.
- **.DmxStreaming**: DreamShow file format that records a DMX stream frame by frame. It is used by the standalone player and by the DMX stream by an AI.
- **Downstage**: the part of the stage closest to the audience.
- **Dynamic effect**: block that varies a setting (intensity, colour, rainbow, pan, tilt, circle, figure eight) according to a waveform, each fixture offset from the others to create a wave, a chase, a fan… See Dynamic effects.

## E

- **Effect**: named setting of one or more fixtures or groups, stored in the library and reusable in all shows.

## F

- **Fade**: gradual change from one lighting state to another. A scene's **fade to the next** is the length of the transition to the next scene, on GO or automatically; a block's **fade in** and **fade**

**out** raise and lower its intensity only. See Fades and mixes.

- **Final state**: the stable lighting of a scene, once its timeline has finished. It is also what you get by jumping directly to the scene. A fixture not set in the final state is at zero (except a winch, which is kept).
- **Fine channel (16-bit)**: second channel that refines a setting (pan, tilt, intensity): the two together give 65,536 steps instead of 256, for smooth movements and fades.
- **Fixture**: a lighting device; DreamShow uses the word for any device patched in the venue (light, moving head, smoke machine, relay…).
- **Fixture definition**: the description of a model: its modes, its channels and what each value does. Imported formats: GDTF, QLC+ (.qxf), DJworld (.fxt).
- **Fixture library**: the set of fixture definitions known to DreamShow (over 1,600, from QLC+ and Polux), which you can add to. See The fixture library.
- **Follow spot**: a light that follows a performer. In DreamShow, moving heads driven by finger from a tablet. See Follow spot on a tablet.
- **Front light**: fixture placed in front of the actors, on the audience side.
- **Full light**: the whole stage lit at maximum.

## G

- **GDTF**: open fixture-description format, supplied by many manufacturers.
- **Go (GO)**: the order to move on to the next scene, by the **GO** button, the space bar or a remote.
- **Gobo**: cut-out pattern placed in front of a moving head's lamp, which projects a shape (star, foliage…).
- **Group**: named set of the venue's fixtures (“Front lights”, “Moving heads”…), which you set all at once.

## H

- **HTP** (“highest takes precedence”): mixing rule where, between two sources, the higher intensity wins. DreamShow applies it to intensity between the cue list and a parallel sequence.
- **Humanoid robot**: Booster K1 robot driven from the timeline by Robot blocks. See Humanoid robots.

## I

- **Invisible**: setting of a venue device that is drawn neither on the plan nor in the simulation (relay, power supply…), but remains patched and driven.
- **IP video player**: video box connected to the network, which plays videos stored on itself, on DreamShow's orders.

## J

- **Jump**: “Go to scene…” block that moves the cue list to another scene when the red line reaches it.

## K

- **Key frame**: recorded step of a trajectory (position and settings of the moving heads at a given moment); DreamShow computes the transition from one key frame to the next.

## L

- **Label**: short, stable name of a scene (for example “rain”); jumps and parallel sequences target the scene by this name, even if scenes are inserted or moved. See Jumps and scene labels.
- **LED mask**: mask with an LED screen, driven over Bluetooth by MaskCtl, which displays faces, pictures or a talking mouth. See LED masks.
- **Libraries** (panel): panel of the main screen that contains the groups, the media, the effects, the trajectories and the **Others** list, to be dragged onto the timeline.
- **LIVE**: button on the main screen; when on, the selected scene is sent to the DMX interfaces; when off, only the simulation carries on.
- **Log**: panel that displays DreamShow's messages (information, warnings, errors). See Log, Audio and Clock.
- **Loop**: setting of a scene whose timeline restarts by itself until GO (or until a maximum length).
- **LTC** (Linear Time Code): timecode carried as an audio signal, on a sound track or an audio cable.
- **LTP** (“latest takes precedence”): mixing rule where the most recent source wins. DreamShow applies it to everything except intensity (colour, position…) between the cue list and a parallel sequence.

## M

- **Manual controls**: window that drives by hand the devices declared “manual” (winch, moving-head power supply, shutters), forced permanently, outside the sequence. See Manual controls.
- **MaskCtl**: program supplied with DreamShow that links the LED masks to the PC over Bluetooth.
- **Media**: a show's sound, video and DMX stream files, stored in the `medias\<show name>` folder.
- **MIDI**: communication standard for musical instruments and controllers; DreamShow accepts MIDI controllers (buttons, faders) and sends the state back to their LEDs. See MIDI controllers.
- **MIDI clock**: tempo signal sent by a MIDI device, which DreamShow can follow.
- **Mix**: on a single track, two light blocks that overlap blend into each other during the overlap: colours and positions glide from one to the other. See Fades and mixes.

- **Mood**: overall lighting setting (full light, day, evening, night, warm interior, cold exterior, mystery, audience in, blackout) used by the wizards and by show creation with an AI.
- **Moving head**: fixture with a motorised head that points remotely (pan and tilt), often with colours, gobos and zoom.
- **MTC** (MIDI Time Code): timecode carried over a MIDI link.

## N

- **No direction**: setting of a venue device (smoke machine, LED strip, house lights…) that lights up with no oriented beam.
- **Node (Art-Net)**: box that receives DMX over the network and outputs it again on DMX sockets.

## O

- **OSC** (Open Sound Control): network control protocol, used by TouchOSC, Companion, QLab… to drive DreamShow. See The OSC remote.

## P

- **Palette**: named colour, position or beam setting, applied with one click; changing the palette changes all the scenes that use it. See Palettes.
- **Pan**: horizontal rotation of a moving head.
- **PAR**: fixed fixture with a wide beam, nowadays most often with colour LEDs.
- **Parallel sequence**: second playback started from a scene, which plays at the same time as the main cue list. See Parallel sequences.
- **Patch**: the assignment of fixtures to their DMX universes and addresses.
- **Pixel mapping**: block that runs an animated picture (gradient, rainbow, wave, fire, image…) over the cells of LED bars, matrices or fixtures. See Pixel mapping.
- **ppvideoplayer**: the video player installed with DreamShow, which plays videos on the monitors.
- **Prism**: optical element of a moving head that multiplies the beam.

## Q

- **QLC+**: free lighting software whose library of definitions (.qxf files) DreamShow reuses.

## R

- **Ready**: pre-show state (blackout, media stopped); the next GO starts scene 1. **Reset (Ready)** brings you back to it.

- **Red line**: the vertical line of the timeline that shows the moment being played.
- **Remote (GO keys)**: 2.4 GHz presentation remote whose buttons send keyboard keys, assigned to GO, previous step and blackout. See The remote (GO keys).

## S

- **sACN** (E1.31): another protocol that carries DMX over a network; universes numbered from 1 to 63,999.
- **Scene**: a step of the show, triggered by GO, automatically or by chaining. It has a timeline and a final state.
- **Screen**: video output declared in the venue: a Windows monitor (played by ppvideoplayer) or an IP video player. See Video screens.
- **Show**: the performance in DreamShow: a sequence of scenes, saved in a `.show.json` file.
- **Show mode**: use in performance, with the cue window. See Show mode and the cue window.
- **Shutter**: fixture channel that opens, closes or flashes (strobe) the beam.
- **Simulation**: the venue drawn in 2D or 3D with its beams and screens, locked to the red line. The ⏻ button switches it off. See 2D and 3D simulation.
- **Slot**: location of an IP video player with which a video file is associated.
- **Sound-to-light**: function where the music (microphone, line input or the PC's sound) varies a group's intensity, with tempo detection. See Sound-to-light and tempo.
- **Stage left**: the right-hand side of the stage seen from the audience (called “cour” in French).
- **Stage light**: stage lighting unit (see also fixture, moving head, PAR, wash).
- **Stage right**: the left-hand side of the stage seen from the audience (called “jardin” in French).
- **Standalone player**: small box (ESP32, Raspberry Pi, mini-PC) that replays an exported show without a PC. See The standalone player.
- **Strobe**: device, or shutter setting, that produces rapid flashes. DreamShow never triggers it automatically from a wizard or an AI.

## T

- **Tap tempo**: giving the tempo by tapping in rhythm on a button (**TAP**) or a key.
- **Tilt**: vertical tilt of a moving head.
- **Timecode**: time signal (hours:minutes:seconds:frames) coming from a sound track, a desk or software, which starts each scene at its time. See Timecode (LTC / MTC).
- **Timeline**: a scene's strip of time, with its tracks and blocks. See The scene timeline.
- **Track**: horizontal line of the timeline that carries blocks. A scene can have several tracks that play at the same time.
- **Trajectory**: movement of moving heads described by key frames, placed on the timeline like a block. See Trajectories.
- **Trigger**: the way a scene moves on to the next: **Manual GO**, **Auto** (after a delay) or **Chained** (when everything it contains has finished).

## U

- **uDMX**: small one-universe USB-DMX interface.
- **Universe**: a set of 512 DMX channels, that is, one DMX line. A large venue uses several.
- **Upstage**: the part of the stage farthest from the audience.
- **USB DMX PRO**: USB-DMX interface (Enttec DMX USB PRO type), recognised by Windows as a COM port.

## V

- **Venue**: the place described in DreamShow: its dimensions, its fixtures placed in metres, their addresses, their groups, its screens, its DMX connections. A venue can serve several shows.
- **Visualiser mode**: mode where DreamShow receives DMX from other software or a lighting desk (Art-Net or sACN) and displays it in its simulation, without sending anything. See Visualiser mode.

## W

- **Wait**: timeline block that stops playback until a key, a length of time, a time of day, a date or sunrise / sunset. See Waits.
- **Wash**: fixture (often a moving head) with a wide, soft beam, which “washes” a surface with light.
- **Web remote**: phone or tablet on the network that drives the show from its browser. See The web remote.

## Z

- **Zone**: tool that draws a zone on the plan; the selected moving heads cover it. In the moods: stage right, centre, stage left, downstage, upstage or the whole stage.
- **Zoom**: setting of the beam's opening, from the tightest to the widest.

## See also

- Welcome to DreamShow
- Main screen overview
- Keyboard shortcuts