DMX I/O

Simple DMX lighting console for macOS with Art-Net & sACN output and MIDI and OSC support for external control.

DMX I/O is a native macOS app written in SwiftUI.

Requirements

- macOS Sequoia (15) or higher
- Apple silicon Mac (Intel not supported)

Overview

DMX I/O is designed to fulfill basic needs for DMX lighting control on macOS. Fixture parameters and raw DMX values can be changed directly in the interface (basic scenarios), or via MIDI messages (easier for more complex setups). You can use for example TouchOSC to create a custom control surface to control your lights.

The app can output universes with DMX values via Ethernet (Art-Net, sACN) or an Enttec USB adapter.

There is a modular philosophy behind the interface. Complexity to your project can be added in steps. From the start you can immediately control your DMX universe with just DMX channel faders. Later, you can patch fixtures for better control, use scenes & presets, work with the command interface and start working with MIDI input from other control surfaces.

This app is not designed to compete with full-featured lighting consoles. It lacks a full FX engine, a large fixture database and other advanced features.

You won't use this app for a stadium show. However, DMX I/O is ideal for situations where you don't want to use a full lighting console. Examples include testing, teaching/learning, school plays, backup for

another light controller or experimenting with DMX512 and Art-Net. You can also use it in production environments for controlling a simple lighting rig or any small project with just a few fixtures.

> ! Caution
>
> Take into account that this is still alpha-quality software; don't use it in mission critical situations.

Architecture & Philosophy

A core architectural choice of DMX IO is to build around direct DMX control. When you change fixture attribute values, you're directly manipulating DMX universe values in a memory buffer. The DMX Buffer is the "Single Source of Truth". This design is sometimes referred to as stateless.

In contrast, major lighting consoles (such as grandMA3 or ETC EOS) utilize a Parameter Abstraction Layer (PAL). A PAL dynamically renders DMX values by calculating a (often complex) priority hierarchy—sorting through effects, cues, and faders in real-time to determine which source takes precedence. This is a stateful design.

With a parameter abstraction layer, if a fixture is controlled by two or more sources, the hierarchy determines which source takes precedence. For example, effects might take priority over faders.

With direct DMX control on the other hand, the last source (for example a cue, or a preset) that sets a certain DMX value, is the source of the current DMX value. There is no layering. This removes a lot of complexity when it comes to determine where a certain output value comes from.

This also means that if you change an attribute value (= DMX channel), the old value is lost. You can not stay "in a cue" for example. Whatever source changes DMX values after a cue was applied, it overwrites values set by the cue. That makes DMX output very predicable: the "last Action" always wins. If you send a command, you see the results form that particular command immediately.

There also is no distinct "Programmer" layer to clear. In DMX I/O, you are essentially always in "Programmer Mode," directly shaping the live output in real-time.

This architecture offers a streamlined workflow similar to QLab lighting control, Lightkey, and the hybrid approach of QLC+, rather than the deep abstraction of a touring-grade desk.

To summarise key distinctions between DMX I/O and a touring-grade console:

- Persistence: You do not "stay" in a cue. A scene (cue) is a command that sets the buffer to a specific state. If a manual fader is moved afterward, it simply overwrites that state.
- Continuous Programmer State: There is no distinct "Programmer" layer to clear. In this app, you are essentially always in "Programmer Mode," directly shaping the live output in real-time.
- Output predictability: Working without a PAL significantly reduces the overhead required to determine why a light is doing what it’s doing—there are no hidden priorities or "tracking" values to troubleshoot.
- Direct DMX control: DMX I/O offers very direct control of fixture DMX channels while also providing control via abstract parameters. Every functionality available in a fixture’s manual, can be accessed by directly manipulating DMX channel values. Even if there is no profile available for a particular fixture. In contrast to major lighting desks where raw DMX values are completely hidden behind an abstraction layer. But, DMX I/O also offers convenient more abstract control by providing a way to use fixture profiles. While DMX IO engine manipulates the DMX buffer directly, it provides a way to use fixture Profiles to translate complex lighting attributes into simple user controls.
- Performance. In professional consoles there is a constant performance impact for synthesizing the universe every frame, often 30–40 times per second. With direct DMX control, there is only a calculation (which DMX channel number to change) needed when a values changes, or when a current value needs to be retrieved. The rest of the time the time DMX processing is idle. Besides rendering the UI, the only constant workload is the sending of the DMX memory buffer at a constant frequency to Art-Net, sACN or USB devices.

For small productions, installations, labs, and AV tech workflows this is ideal. But, is is not a touring desk architecture.

Basic setup

Main screen

The main screen contains several views, such as channel faders, fixtures and scenes. You can hide views that you don't need in the "View" menu.

In the top bar of the main screen you'll see these controls:

- Universe selector. This determines the universe for the DMX channel faders.
- "Blackout" button (shortcut: `CTRL` + `C`). This button sets all DMX channels to `0` and stops running effects. This can be used as a "Panic" button. It is also a quick way to start building a scene or preset from scratch.

Show file

Scenes, fixture patch, presets, etc, can be saved and loaded from a file (called a Show file) via the File menu

> ⚑ Important
>
> Don't forget to save your changes to a file before you close the app. There is no auto-save functionality.

Basic Control: DMX channel Fader view

This view contains faders for all DMX channels in the selected universe, starting with channel 1 (left). If a fixture is assigned to the channel, the channel parameter name is shown in the fader.

- The text color above each fader indicates the source of the value:
  - gray: not set
  - green: value was changed manually, that means in UI via fader, by running a command, or received via MIDI
  - yellow: value was received via Art-Net input
  - white: highlight mode on for this channel
  - purple: value is from an applied preset
- The number in the fader is the current DMX value. Use `CMD` + Left click to toggle between absolute value and percentage.

Patching, fixtures & profiles

Patching

Setting up fixtures and their DMX addresses via the Patch Editor in the app opens up additional functionalities, like easily controlling color attributes via a color picker.

The Patch editor can be opened from the Window menu. In this editor, you can patch fixtures: assign DMX addresses, select profiles, and configure other fixture properties.

Notes about fixture properties:

- Fixture ID - You can manually assign a numeric ID to a fixture. Similar to a channel number in ETC consoles. Fixtures in the pillbox view are sorted by this ID. An often used numbering scheme is to use a new 100-level for each fixture type or location. Organising fixtures based on IDs can be done in several ways. Some examples:
    - By type:
        - Spots: 101-105
        - Washes: 200-205
        - Strobes: 301-310
    - By location:
        - Front lights: 101-105
        - Back truss: 200-205
    - Both Type/location:
        - Front wash: 101-111
- DMX Address This value is the DMX start channel of the fixture within its assigned universe. This is a one-based DMX address (1-512). Fixture channels cannot cross universes.
- Color - This is the display color shown in the pillbox tile for the fixture.
- Label - This is the text label shown in the middle of the pillbox tile for the fixture. It's a way to assign a "friendly" (easily recognisable for you) name to a fixture.
- Tags - By assigning free-form tag names to a fixture, you can group similar lights. All fixtures that have the same tag can be triggered or selected together. Tags can be used as shortcuts to quickly select fixtures. You can also assign a MIDI Controller number (0..127) to a tag. This number can be used to control all fixtures with that tag via MIDI.

A fixture patch is saved in the show file.

Profiles

The app ships with a limited number of fixture profiles built-in.

You can create your own profiles by writing JSON fixture profiles. Such user created profiles are automatically loaded from the following directory on app startup:

```
~/Library/Application Support/eu.onderweg.DMXIO/profiles/
```

Example of how a profile JSON should look:

```
{
    "awesome_led" : {
        "id": "<unique id, preferably a UUID>",
        "name": "LED",
        "mode": "standard mode",
        "manufacturer": "Awesome",
        "parameters" : [
            {
                "offset" : 0,
                "type" : "dim"
            },
            {
                "offset" : 1,
                "type" : "red"
            },
            {
                "offset" : 2,
                "type" : "green"
            },
            {
                "offset" : 3,
                "type" : "blue",
                "name" : "optional name for parameter",
                "defaultValue": 50,
                "highlightValue": 128
            },
            {
                "offset" : 4,
                "type" : "tilt"
            },
            {
                "offset" : 5,
                "type" : "tilt_fine"
            }
        ]
    }
}
```

Notes about creating profiles:

- Multiple profiles can be stored in one JSON file.
- Create a separate profile object for each fixture mode (personality). Use the `mode` attribute, containing a mode name, to differentiate between different modes of the same fixture.
- Every profile should have a unique key amongst all fixtures (`awesome_led` in the above example). If there are multiple profiles for the same fixture, one for every mode (also sometimes called personalities) of a fixture, every mode should have its own unique key. Consider using a `GUID` to guarantee uniqueness.
- A parameter in a DMX I/O profile is mapped to an 8-bit DMX channel. You can map a 16-bit parameter by creating a coarse and a fine channel (postfix `_fine`) in the fixture profile. E.g. to control a 16-bit red channel, create a `tilt` and `tilt_fine` parameter next to each other. See JSON example above.
- A list of all supported parameter type names can be found in menu Window -> Fixture Parameters.
- In the JSON fixture parameter object, `offset` and `type` are required. Others like `name`, `highlightValue` and `defaultValue` are optional.
- A fixture mode can contain multiple attributes of the same type. For example, some fixtures have more than one `macro` or `effect` channel. Although having multiple attributes of same type name in a profile is supported, controlling them can be tricky. For one thing, presets only store one value per attribute type. Therefore, for some attributes there are numbered versions available. For example, for a channel of type `macro`, you can use `macro2` for the second macro channel. This is the recommended way to define a fixture with same-type channels.

> There is no direct support for multi head/cell/beam fixtures, such as Chauvet Strike M, which has DMX channels for multiple plates/beams. Workaround: Create a fixture profile for each cell/head/beam that you want to control.

Patching "Generic" Fixtures

Commonly you will encounter fixtures that may not be in the fixture library, but can operate with "generic" modes - i.e. a 3-channel RGB fixture.

These fixtures are categorized under the "Generic" manufacturer.

Patching Conventionals

To patch conventional dimmers and fixtures, select "Dimmer" fixture profile from manufacturer "Generic".

Fixture Tags

Using tags is a way to group similar lights and control them together. Every fixture can have zero or more tags. Tag names can be chosen freely. All fixtures that have the same tag can be triggered or controlled together.

> 💡 Tip
>
> With the tags view (menu "View" -> "Show tags") you can quickly select all fixtures belonging to a tag.

Fixtures view (pillbox)

In Pillbox View, each tile represents a patched fixture. In this view, you can quickly set parameters for multiple fixtures by selecting them and sending MIDI commands, or using the fixture controls accessible via the button on top of the main window.

In each tile ("pillbox"), the following information is shown:

- Fixture DMX start and end address (top), or fixture id (configurable)
- Fixture label (middle)
- Current intensity percentage (bottom)

You can select one or more fixtures in the pillbox view by clicking on them. A yellow border will appear. A selection can be used to control the same parameters and apply presets to a set of fixtures.

Scenes (Cues) and scene view

Scenes - also called Cues, Looks, etc. in other lighting software - are stored universe states. A scene can store DMX channel values of entire universes or of partial universes (one or more channels).

In contrast to major lighting consoles like GrandMA and ETC, scenes in DMX I/O are not stored as a set of commands, presets or parameter values. DMX I/O stores scenes as a snapshot of one or more raw DMX channels.

> 💡 Tip
>
> The approach of storing scenes as raw DMX values has some advantages but also several downsides.
>
> Because a scene in DMX I/O is just a snapshot of raw DMX channel values, you can even work with scenes if you did not patch any fixtures. Also, when you store a scene, what you see is what you get: the exact current stage look is captured, and the exact DMX output that produces the look can be played back later.
>
> A downside is that a scene does not reference abstract data, like presets. You might have to re-record scenes whenever you change (e.g. add/remove fixtures).

A scene can be assigned a custom number. When you edit a scene, you can change this number to any number that is a positive number (integer). The scene number does not determine the sorting order; it cannot be used to move the scenes around.

Scene numbers and ordering

Scenes do have an order, that is independent of scene number (scene with number `3` can occur before scene `9`.). Scene tiles are displayed in that order. The order is also used when a next/previous scene command is given. To re-order scenes: press the "Reorder" button below the scene tiles to show the reorder dialog, and drag scenes to reorder them.

You can assign a scene an external identifier: a free-form string like `2.1.3` in addition to the scene number (which is always an integer).

This is useful for example when you need to reference a scene defined in an external system like QLab, or a scene from a script.

To achieve this, you can fill in the "external_identifier" field when you edit a scene. An external identifier is a string value (can be anything). This can be used as an alias to reference the scene.

Scenes have optional "next scene number" (sometimes called "follow" in other lighting software). This can be used to link multiple scenes together for non-linear playback.

Scene tiles

Scenes are shown in the scene view as square tiles. The following information is displayed in a scene tile:

- The number at the top left is the scene number
- If set, at the top right the external cue number is shown. An external cue number can be used to reference the cue from an external control system via MIDI or OSC.
- The number at the bottom represents the number of DMX channels recorded for that scene. One full universe is `512` channels.

Selecting scenes

Select (also called "load") a scene by clicking on it. This does not activate a scene. `Command` + click will select and immediately activate the scene.

The selected scene is the scene that will be played when the Go or Play button is pressed, or recorded into when the Record button is pressed.

A selected scene in DMX I/O is comparable to:

- The cue with the playhead in QLab.
- pending cue in ETC.
- The standby position in MIDI Show Control protocol.

Controlling scenes

Recording

Recording a scene essentially takes a "snapshot" of DMX channels in all universes. If you click on the "record" button, you can choose to record only certain channels.

- A use case for recording all DMX channels in a universe is creating a "Blocking Cue." If you want to ensure that no values from a previous cue track through into this one

> ⓘ Note
>
> Recording channels into a scene works like a merge: if there is an existing value for a channel already in the scene, that value is replaced by a new one. Other existing channel values in the scene are left untouched.
>
> To start with a clean slate instead: clear the scene first.

Playing

Playing (activating) a scene means that DMX channels in the universes are set to the values stored in that scene.

> ⚑ Important
>
> DMX I/O does not have the concept of being "In" a scene. When you activate a scene, DMX channels for that scene are set. Other channels are not affected by that scene. What you see on stage can be a combination of multiple scenes, presets, manual values, etc. See also how QLab works with cues, which is similar.
>
> You can see the latest run scene in the row with buttons below the scene tiles.

Below the scene tile view, there are buttons to activate (play), record (store), rename, and add scenes. These scene control buttons apply to the currently selected scene.

- Modify the scene by pressing the "Edit" button.
- Record values for the scene, by pressing the "Record" button
- Play the selected scene:
  - By pressing the "Go" button or pressing `SPACE`, to play the selected scene and select the next scene. The Go button functions as the Go command in QLab. It will play the selected scene and advance the selection to the next scene.
  - By pressing the "Play" button, or pressing `CMD` + `P`, to play the selected scene.
  - By holding `Shift` whilst pressing "Play" to snap to a cue (for example, during a programming session when you want to alter a cue without having to wait for the cue fade to complete).

DMX I/O uses the principle of tracking. When a value is set in a scene, that value continues (is "tracked") into later scenes. The universe will maintain that value in all subsequent scenes until it is explicitly told to change.

If a scene is selected, you can start a fade-out of intensity channels. The fade-out duration can be set in the scene edit dialog. Two fade-out buttons are available:

- Fade-out intensity channels in selected scene: only intensity channels that are part of the selected scene (and belong to a patched fixture) will fade out. Before fade out starts, this sets the scene channel values to intensities as stored in the scene. After fade-out, the next scene is selected.
- Fade-out all intensity channels: use this button to fade out all intensity channels of all fixtures. This is a complete fade-out to black.

There are buttons available for selecting the next and previous scene.

Advanced scene control (sequences)

In a theatrical setting, you might need more advanced cue control. Like automatically holding a cue for a set amount of time before proceeding

to the next, or follow, where the cue goes right after the previous cue finished.

Executing a workflow with advanced cue sequencing can become very complex quickly. The philosophy behind DMX I/O is that this can be better handled by dedicated cueing software like QLab.

DMX I/O provides multiple ways to trigger scenes easily by other show control software, via: MIDI voice messages (e.g. Program Change), MIDI Show Control messages and OSC. See section External Control for more information.

Scene transitions (experimental)

Optionally you can set a scene fade in/out times when editing a scene.

- When a fade in time is set, that time is used to fade in intensity channels of the scene when you play the scene. Fade in starts from 0, so if scene channels have already set intensity values, these are first set to 0 before fade in.
- When a fade out time is set, that time is used to fade out intensity channels of the current selected scene when you do a scene intensity fade out. Fade out works with what is known in ETC as "assert": before fade out the entire state for scene channels is sent out, fade out starts from there.
- If the fade in or -out is `0` seconds, scene values are set immediately and no fading occurs.

Scene live list window

From the Window menu, you can open a "Scene Live List" window. This opens a window with the scene list. This can be convenient to have the scene list always visible on a separate display for example.

A red triangle indicates the playhead, it points to the scene that will play on Go. This is the same as the currently selected scene in the scene tiles. Keyboard shortcuts in this view:

- `space` Go: play scene which is currently standing by at the playhead and move playhead to next scene.
- ↑, ↓ Move scene selector up/down, press `enter` to load scene (set playhead to selected scene).
- `CMD` + `F` Fade out intensity value of scene standing by

Presets

A preset contains saved values for a specific attribute group, such as color, position, intensity, or beam. In some lighting consoles these are called Palettes.

> 💡 Tip
>
> Presets versus scenes.
>
> Presets are mainly used to build scenes during programming. They are dependent on the selection of fixtures in the pillbox view.
>
> Scenes are used for playback during a show. They capture a full look on stage at a point in time. As scenes store DMX channel levels, they don't depend on selected fixtures. Scenes can be built conveniently with the help of presets.
>
> You can think of presets as building blocks during programming, and scenes as stage looks during a show.

Preset view

All stored presets are shown in the Preset View, which can be shown/hidden from the View menu.

- Left clicking on a preset will apply the preset values to currently selected fixtures.
- Right clicking on a preset will show a context menu with options to modify the preset.

Preset groups

Every preset is assigned to a preset group. The group determines the type of attributes stored.

Currently the following preset groups are built-in:

- Color. Stores RGBW color information that can be applied to selected fixtures. This can be used for example to create color palette presets.
- Position. Stores pan/tilt attributes.
- Dimmer. Stores intensity values.
- Custom. Stores any combination of values.

> 💡 Tip
>
> Besides the predefined preset groups, you can store any arbitrary set of attributes in a custom preset.
>
> Use the command `record custom` to create a custom preset from the attribute values of currently selected fixtures. For example, to store `strobe` and `gobo` attributes in a preset:
>
> ```
> record custom "my preset" strobe gobo
> ```

While scenes save absolute DMX channel values, a preset stores values for specific fixture attributes. Since scenes are channel based, after changing a fixture patch (DMX addresses of fixtures), you might need to re-record your scenes, whereas a (universal) preset remains applicable after a patch has changed.

Presets store attribute data in an abstracted representation. For example, colors are stored as `0.0..1.0` normalized RGB values. Actual DMX values differ per fixture and are calculated when applying the preset. In this way you can use for example the same color preset for both a CMY LED fixture and an RGB fixture.

> ⚑ Important
>
> Color wheels in color presets

> Color wheel values are not stored in color presets. Because color presets are designed to store abstract color data (RGB(W) values) that can potentially be applied to any colour mixing (CMY/RGB/RGBW) fixture. That is in contrast with color wheels, who are very specific to a fixture type.
>
> To store color wheel value in a preset, use custom preset instead.

Fixtures with recurring parameters

If multiple parameters of a type are present in a fixture, for example multiple "dim" parameters in a sunstrip which has multiple lamps, then only the value of the first parameter of its type is stored in the preset.

In the Sunstrip case for example, only the intensity of the first lamp is stored. But, when you apply the preset, you can apply to all parameters of that type. For example, if you stored intensity `25%` in a preset, you can set all Sunstrip lamps to intensity `25%` at once. You can do that by editing a preset, and selecting to which attributes the values apply (first, or all).

For better control, use numbered attributes in your profile, for example `color` for first color channel and `color2` for second color channel (not supported for all attribute types). If you create a preset in that case, both values for `color` and `color2` will be stored.

Preset modes

DMX I/O supports three types (modes) for presets:

- Universal presets, apply the stored values to all fixtures with the relevant attributes.
- Global presets, apply the stored values to all fixtures of the same type (profile). The profile is stored as part of the preset.
- Selective presets, apply the stored values to a stored selection of fixtures. The fixture selection is stored as part of the preset. Therefore, to apply a selective preset, there is no fixture selection needed. This makes it a very fast one-touch button preset.

> 💡 Tip

> Preset mode names in DMX I/O follow the same naming and principle as in the GrandMA console software.

Because global and selective presets are based on specific fixtures, you can only record a preset in this mode if you have selected one or more fixtures.

Universal presets can be very powerful, because they are fixture agnostic. Meaning that even if you completely change your patch, all universal presets still work unaltered.

Creating a preset

To create a new preset:

- Preparation:
  - Select fixtures in pillbox view
  - Set attribute values to the selected fixtures. For example, set colors and intensity levels.
- To create the actual preset, either:
  - UI: Click on "Create preset" button in presets view.
  - Commands: Enter one of the following commands to create a universal preset:
    - Color preset: `record color "<preset name>"`
    - Position preset: `record position "<preset name>"`
    - Dimmer preset: `record dimmer "<preset name>"`
    - Custom preset, with custom set of attributes: `record custom "my preset" strobe gobo`

A preset is assigned a number. When you edit a preset, you can change the preset number to any number that makes sense to you, as long as it is a positive integer. Preset identifiers like `1.2.1` are not allowed.

> ⚑ Important
>
> If you selected multiple fixtures when creating a Universal preset, there can be fixtures who have different values for the same attribute. In that case a "last takes precedence" approach is used:

> the value from the fixture with the highest fixture ID is stored as universal value.

Applying a preset

- Select fixtures
- Apply preset. Either via:
  - Command: `preset <preset number>`
  - UI: click on desired preset. Preset is immediately applied.

Deleting a preset

- Either:
  - UI: Right click on preset in preset view, and choose "Delete preset"
  - Command: Enter the following command: `delete preset <preset number>`

Edit/Delete/Replace/Reorder a preset

Right click on a preset and select an option from the context menu.

Saving presets

Presets are saved on disk in the show file.

> ⚑ Important
>
> Always make sure you save any changes in presets in your show file before you exit the app

Formulas (preset sets)

A formula is a group of universal presets. When you apply a formula, all presets in the formula will be applied to the selected fixtures.

While a universal preset stores a specific set of attributes, a formula stores attributes across multiple attribute families, for example color, intensity and position attributes.

> 💡 Tip
>
> Typically you would apply a formula to a group of the same type of fixtures, for example fixtures tagged "Front Wash".

This can be a powerful programming tool. Because even if you clear your patch, and patch different fixtures, your formulas still work (as long as the new fixtures have matching attributes). With the help of earlier defined formulas, you can now quickly (re)build scenes.

- To create a formula: right click on a universal preset, in the context menu select "Add to formula..." -> "New formula". Formulas will appear below the presets.
- To add a preset to a formula: right click on a universal preset, in the context menu select "Add to formula..."
- To apply a formula: click on the formula.

Chase (experimental)

You can use a chase to create a dynamic lighting effect where lights turn on and off in a programmed sequence. Every step sets lighting parameter values for a set of fixtures.

Steps are stored as a timed sequence of presets for color, intensity and position parameters, for a selection of fixtures. So basically a chase in DMX I/O is an animated series of selective presets.

Chase view

The chase view contains a tile for every chase you created.

The chase tiles show:

- Chase name
- Number of steps in chase
- Chase number

Clicking on a chase will start the chase. When you click on an already playing chase, it will stop.

Creating a chase

Creating a chase from scratch:

1. Create a new empty chase: click "Create chase" in the Chase view.
2. Add steps. To add a new step to a chase:
    1. Select fixtures to include in this step
    2. Set fixture values for selected fixtures. For example set selected fixtures to a certain color.
    3. Record step: right click on chase, choose "Record new step" in the context menu. This will record current parameter values for intensity, color and position of the selected fixtures.

Creating a chase from an existing selective preset:

1. Right click on a selective preset to show the context menu
2. Select "Add as chaser step to..." to copy values from this preset into a chaser.

Changing chase steps

To change steps (edit timing, etc.):

- Right click on a chase, choose "Steps..." in the context menu. Here you can preview/delete steps and change timing.
- Every step has a "Duration" value. This is the time before a running chase moves to the next step.
- For every step, you can select a fixture filter. By default, there is no filter: the step is applied to all fixtures that were selected when the step was recorded. Other filter types:
    - random: every time this step runs, it is applied to a random selection of the fixtures stored in the step.
    - sequence: in every iteration of the chaser effect, the step is applied to the next fixture in sequence.

Common workflows/recipes

This section contains some how-to's for common things you want to achieve with your lighting setup. You may notice that often things can be achieved in multiple ways.

Bump

A Bump (flash) group of lights / Blinder effect can be achieved in multiple ways:

- Using a MIDI trigger to set intensity values for a fixture group. Works nicely with external (MIDI) controllers.
  - Send MIDI to set full intensity to start bump
  - Send MIDI to set 0 intensity to end bump
- Using an intensity preset in the UI by clicking & holding. Ideal for manual control without external controller.
  - Create a dimmer (intensity) preset for a selection of fixtures.
  - `CMD` + Click on the preset and hold mouse/touch down to apply the intensity preset while mouse down. This works best if you created a selective preset, because then you don't have to select fixtures first when applying.
  - Release click to release preset.
- Using an intensity preset with hold time. This is ideal for a fixed-length bump.
  - Create a dimmer (intensity) preset
  - Edit preset to set a hold time
  - To bump: select fixtures, activate preset. This will set the intensity of selected fixtures to the value in the preset, after the hold time, intensity will go automatically back to previous value

Move in Black (MIB)

- Before that cue that requires a certain prositon, create a cue with intensity channels `0` and position channels that have the end value.

Commands and commands view

Commands view

In this view, you can quickly control several aspects of your lighting rig by typing simple commands. Most commands are applied to selected fixtures in the pillbox view.

Selections

The commands select one or more fixtures in the pillbox view.

```
# Select fixtures 601, 602, 603, 604
601 Thru 604
# Select fixture 100, 115 and 120
100 + 115 + 120
```

`at` commands

These commands set parameter/channel values.

The basic command syntax for this type of command is: `<parameter> at <value>`.

Examples:

```
# Set dimmer channels of all the selected fixtures to value
        30%:
dim at 30
# Set position channels
tilt at 50
pan at 50
```

A fixture can have multiple parameters of same type. For example multiple dimmer channels. The commands above change all dimmer channels.

To change only the 2nd dimmer channel of all the selected fixtures to `30`:

```
dim.2 at 30
```

The list of parameters available for a specific fixture can be found in the Patch editor window.

Scenes

```
# Go: Play scene 5
go 5
# Select, don't play, scene 5
scene 5
```

Presets

```
# Record universal color preset from selected fixtures
record color "my preset"
# Record universal dim preset from selected fixtures
record dimmer "my preset"
# Record universal preset, with custom set of attributes
record custom "my preset" strobe gobo
# Apply preset
preset <preset number>
# delete preset
delete preset <preset number>
```

Auto-generate common presets:

```
auto presets
```

Formulas

```
# Create new empty formula, with number 4
create formula "my formula" 4
# Apply formula 4 to selected fixtures
formula 4
```

Raw DMX

DMX values per fixture:

```
# Copy all DMX values/parameters from fixture with id 102
        to fixture 104
copy 102 104

# Clear (set channel values to 0) selected fixtures
clear
```

DMX values in universe:

```
# Set DMX channel 4 of current selected universe, to value
        (level) 120
channel 4 120
# Set intensity DMX channel of current selected fixtures
        value (level) 50
dim at 50 DMX
# Set 16 bits DMX channel
dim at 50 DMX16
```

Other commands

Highlight fixtures:

```
# Highlight selected fixture, so that you can easily locate
        them in your rig
highlight on
# Highlight off
highlight off
# 'hi' is an alias for 'highlight'
hi on
```

Fixture control

```
# Set all parameters of selected fixtures to their defaults
home
```

RGB:

```
# Set red, green, blue component values (0..255) of
        selected fixtures
rgb 5 10 10
# Set RGB values of selected fixtures, with a cross fade
        animation that has a duration of 2s.
rgb 115 115 115 2.0
```

Tags:

```
# Add tag to selected fixtures
tag "my tag"
# Select fixtures with specified tag
tagged "my tag"
```

Macros: learn a sequence of commands and store those under a number.

```
# Start learning
# Commands following this learn command will be stored as
        macro sequence
learn <number>
# Stop learn (ends macro sequence)
learn
# Play stored macro
macro <number>
```

External control

DMX I/O supports MIDI and OSC as protocols for external control

> 💡 Tip
>
> OSC versus MIDI: when to use which?
>
> In general, the advantage of MIDI is that it is very easy to get working. It usually "just works" locally with a controller directly connected to your computer via USB. On macOS there is no need for any setup or installing drivers.

> Downside is that most devices only support MIDI 1, which means parameter values can be sent only in 7-bit resolution (value range `1..127`).
>
> While possible, using MIDI over a network is often less straightforward than using it over a cable.
>
> That's where OSC shines: it's inherently designed to be used over a network. Making it ideal to remotely control an app. For example you can control DMX I/O on macOS with TouchOSC on iPad. Also OSC is much more flexible in types of values that can be sent.

OSC external control

DMX I/O runs an Open Sound Control (OSC) server that listens on UDP port `9011`.

Some use cases for OSC Control:

- Trigger scenes from Streamdeck
- Control parameter values with TouchOSC
- Activate a scene from QLab or Ontime.

Supported incoming OSC commands:

```
# Activate scene by number
# Example: /scene/go 3
/scene/go <scene_number: Int>

# Activate scene by external id
# Example: /scene/go "1.2"
/scene/go <external_id: String>

# Activate selected scene and select next scene.
# Mimics QLab Go behaviour.
# (notice that there are no arguments)
/scene/go

# Load a scene by number or external number
# Selects scene that will be triggered by next Go command
/scene/load <scene_number: Int>
/scene/load <external_id: String>

# Set parameter of selected fixtures
# Use case: select fixtures, change values easily with
external controller
# Example: /parameter/dim 0.3
/parameter/<parameter_name> <normalised_value: Double>

# Set parameter of fixtures with the string tag name
provided as first argument
# Example: /parameter/dim "my tag" 0.3
/parameter/<parameter_name> <tag_name: String>
<normalised_value: Double>

# Set parameter of fixtures with the integer tag MIDI
number provided as first argument
# Example: /parameter/dim 5 0.3
/parameter/<parameter_name> <tag_number: Integer>
<normalised_value: Double>

# Change RGB values of tagged fixtures
# Example: /rgb "my tag" 0.2 0.3 0.4
/rgb <tag_name: String> <red: Double> <green: Double>
<blue: Double>

# Change Position (pan/tilt) values of tagged fixtures
# Example: /position "my tag" 0.2 0.3
/position <tag_name: String> <x: Double> <y: Double>
```

MIDI external control

MIDI Input

DMX I/O exposes a virtual MIDI input device called `DMX I/O MIDI In`. Use this virtual MIDI device to send MIDI commands to control lighting via DMX I/O.

Accepted MIDI input messages:

- Control Change (CC). Used for adjusting fader/parameter levels.
    - `Channel 1` => Changes a parameter value on all selected fixtures in the pillbox view. Controller number determines the parameter type (e.g., intensity). The MIDI value (`0` .. `127`) determines the channel value.
    - `Channel 2` => Changes intensity channel of current/next scene list. Can be use to create a master fader for scenes.
        - `CC 1` : Controls the intensity of the current selected scene. Value (`0..127`) is target intensity.

          `CC 2` Controls the intensity of the next scene in order. Value (`0..127`) is target intensity.
    - `Channel 3` => Changes intensity channels of specific scene number. This can be used to create a scene fader, sometimes called cue submaster or executor. Controller number is used as scene number (`0` .. `127`). When value is zero, the scene is off, meaning all channels that are part of the scene are set to zero. Any value above zero will activate the scene and simultaneously adjust the scene's intensity channels. `127` is full intensity.
    - `Channel 4` => Changes intensity of fixtures having a specific tag (a fixture group). This can be used to create a group master. You can assign a (MIDI CC) number to a tag. That number is used to determine for which tagged fixtures to increase intensity. The MIDI value determines the intensity. You can use this to create a fader for a fixture group on a MIDI controller like TouchOSC.
- Note On. Set raw DMX values.
    - (any channel) => changes a single DMX channel value. The note number is used as DMX Channel number, and velocity as value for the channel. Note that MIDI values range from `0..127` , while DMX channel values have a range of `0..255` . Therefore, received MIDI values are converted to 8-bit DMX channel values by multiplying them by 2.

- Program Change (PC). Used for activating scenes, presets and chases.
    - `Channel 1` => Select and activate a scene by scene number. Program number is scene number to activate. This only works for scene numbers in the MIDI program number range: `0..127`. If you want more extensive numbering, like "1.2.5": set an "External cue number" for the scene, and use MIDI Show Control messages (see below) to start/trigger a scene.
    - `Channel 2` => Scene control (Go/Next/Prev)
        - `Program 1`: Go: Activate selected scene and select next one
        - `Program 2`: Standby +: Select (don't activate) next scene
        - `Program 3`: Standby -: Select (don't activate) previous scene
    - `Channel 3` => activate a preset. Program number is preset number to activate.
    - `Channel 4` => toggle a chase. Program number is chase number to activate.
- MIDI Show Control (MSC). DMX I/O Interpretation of MSC commands:
    - `Go` => Select and activate a scene with the provided external cue number. Can be used to trigger a DMX I/O scene from show control software like QLab. When no Cue number is provided, the currently selected scene is activated, then the next scene is selected.
    - `STANDBY+` / `STANDBY-` => Select (don't activate) next/previous scene.
    - `LOAD` => Select scene with the provided external scene number (required).
    - `ALL_OFF` => Blackout. All channels to `0` and stops all running FXs.
    - All other MSC commands are ignored.

MIDI Output

DMX I/O exposes a virtual MIDI output device called `DMX I/O MIDI Out`.

These MIDI messages are sent:

- Program Change (PC)
  - `Channel 1` => Scene activated. Program number is scene number activated.
  - `Channel 2` => Preset applied. Program number is preset number applied.
- MIDI Show Control (MSC)
  - `Go` command => Sent when scene was activated. This can be used to automatically trigger scenes in other software like QLab from DMX I/O.

DMX Output and input

Art-Net

Each Art-Net device needs an IP address with in the `2.x.y.z` or `10.x.y.z` range. Each DMX universe must then be allocated to a Art-Net Universe (Port). Ports numerate from 0-15, so it is generally accepted that DMX I/O universe 1 will become Art-Net universe 0, however this is user definable.

Art-Net output

The app supports Art-Net output. You need to configure the following settings for Art-Net in the settings dialog:

- The IP address to send Art-Net packets to. This can be a unicast (recommended) or broadcast address (not recommended).
  - When using broadcast, use Directed Broadcast addresses `2.255.255.255` or `10.255.255.255`
- Frame rate (fps). Maximum 44 fps.
- Art-Net universe. Art-Net universe numbering starts from 0.

Art-Net input

The app can receive Art-Net data and merge it into the current universe.

To enable Art-Net input: Menu "Options" -> "Enable Art-Net input"

Use cases for Art-Net input:

- Record DMX output from a console into a scene. You can use this as backup for the main console.
- Monitor DMX output from a console.
- Control DMX values in DMX I/O from another tool.

> 💡 Tip
>
> Art-Net input/output can be easily tested with QLC+.

sACN (E1.31) output

DMX I/O supports streaming ACN (sACN or ANSI E1.31) output. Besides DMX data, it sends universe discovery frames approximately every 10 seconds containing all the universes enabled for sACN output in DMX I/O.

You need to configure the following settings for sACN in the settings dialog:

- Addressing mode:
  - Unicast. For unicast, you must provide the IP address to send sACN packets directly to. Note that most sACN systems default to sending data via multicast.
  - Multicast. For multicast, no destination IP needs to be set. But you should select the network interface to send multicast packets from. Otherwise, packets are sent only out of the default adapter, which is not always desired. For example, if you want to use a wired Ethernet adapter for sACN transmission instead of your default WiFi adapter.
- Frame rate (fps). Typically around `25` to `30` fps. Maximum 44 fps.
- Select which universes are enabled for sACN output

> 💡 Tip
>
> Enable IGMP Snooping on your switch if using multicast for the best performance.

Enttec USB output (experimental)

DMX I/O can output the DMX universe to an Enttec USB Pro or Open USB adapter.

Enable Enttec output in the "Options" menu. The app will automatically detect connected and supported Enttec devices (based on USB vendor and product ID), and connect to the first available device.

Take into account that using a USB DMX adapter is less reliable and stable than outputting DMX over a network (Art-Net, or sACN):

- When device is disconnected, sometimes the app needs to restart to be able to connect again.
- USB port can be blocked by other applications.

> ⚠ Warning
>
> Enttec USB support in DMX I/O is experimental and finicky. It can be safely used for playing around with DMX output. Don't use it in a high-stakes environment. Use Art-Net or sACN output instead.

- Enttec Open DMX devices don't include a microprocessor to create the DMX stream. Instead they rely on the computer to send DMX frames at regular intervals. If the computer can't provide the DMX data at a high enough speed, you may experience flicker. In this case you should try to reduce the DMX refresh rate.

Settings

Use the settings dialog to change application settings and enter your license key.

Settings will be in a JSON file at this location:

```
~/Library/Application Support/eu.onderweg.DMXIO
```

Keyboard shortcuts

Main window shortcuts:

Global controls

- `CMD` + `0`...`9`: change active universe
- `CTRL` + `C`: blackout (all channels to `0`)

Scene controls

- `CMD` + `P`: play selected scene
- `SPACE`: Go (play selected scene and select next)
- ←, → Select next/previous scene

Known issues

- Enttec USB support unstable

Disclaimer

This software is provided "as is," and you use the software at your own risk. The author makes no warranties as to performance, merchantability, fitness for a particular purpose, or any other warranties whether expressed or implied. No oral or written communication from or information provided by the author shall create a warranty. Under no circumstances shall the author be liable for direct, indirect, special, incidental, or consequential damages resulting from the use, misuse, or inability to use this software, even if the author has been advised of the possibility of such damages.
