Merge modes

Fixture merge modes determine which channels (i.e. functions) of a fixture are controlled by the zactrack server during tracking mode and how much (if at all) they are influenced by input received from the lighting console.

Change the merge mode of fixtures in the Fixture Live View. See Create a custom fixture type for information on different channel functions of fixture types.

Note: These settings only affect normal tracking mode. You will always be able to change the four channels DIMMER/IRIS/ZOOM/FOCUS (if available) while performing fixture alignments or setting up autofunctions.

Auto 2

In **Auto 2** mode, the zactrack server only adjusts the **PAN** and **TILT** channels of a fixture according to the actor's position during tracking mode. Other channels (including DIMMER, IRIS, ZOOM and FOCUS) have values according to the Base Ch. Values setting in the fixture live view.

Note that if you defined autofunctions for a fixture, they won't have any effect at all in Auto 2 mode.

Auto 6

In **Auto 6** mode, the zactrack server adjusts **PAN**, **TILT**, **DIMMER**, **IRIS**, **FOCUS** and **ZOOM** channels (if available) of a fixture while tracking an actor. Just like in Auto 2 mode, PAN and TILT values are determined by the actor's position. DIMMER, IRIS, FOCUS and ZOOM channels are set according to the fixture's autofunctions of the currently active scene.

If the smart zoom feature is enabled for an actor, the ZOOM and IRIS channels will be set automatically, regardless of any autofunctions.

If there are no autofunctions defined for this fixture (and the smart zoom feature is disabled), the channels have the following default values:

- **DIMMER**: Full (e.g. DMX value of 255 for 8-bit or 65535 for 16-bit channels)

- **IRIS**: Half (e.g. DMX value of 127 for 8-bit or 32767 for 16-bit channels)

- **FOCUS**: Half (e.g. DMX value of 127 for 8-bit or 32767 for 16-bit channels)

- **ZOOM**: Half (e.g. DMX value of 127 for 8-bit or 32767 for 16-bit channels)

All other channels have values according to the Base Ch. Values setting in the fixture live view.

Auto 6M

Auto 6M mode works exactly like Auto 6 mode in that **PAN**, **TILT**, **DIMMER**, **IRIS**, **FOCUS** and **ZOOM** channels are determined by an actor's position, the fixture's active autofunctions and a potentially enabled smart zoom. However, this mode allows the lighting console to have final control over the four channels **DIMMER**, **IRIS**, **FOCUS** and **ZOOM**.

The zactrack application first calculates base values for these four channels according to the currently active autofunctions smart zoom settings, just like in Auto 6 mode. Before sending these values to the fixtures, however, the application takes the channel values received on a fixture's **input universe** (if defined) and uses them as **multipliers** in order to calculate the final output value.

All other channels have values according to the Base Ch. Values setting in the fixture live view.

Auto 6M Use Full Console Range

There are two different ways of how the multipliers are calculated, depending on whether or not the advanced property in the System Settings is enabled or not.

Option 1: Property "Auto 6M Use Full Channel Range" is enabled:

Let **auto_value** be the value calculated based on autofunctions/smart zoom and **console_value** the one received from the console on the fixture's input universe. Furthermore, let **dmx_full_range_min** and **dmx_full_range_max** be the theoretical dmx value limits, **regardless of the limits set in the fixture's fixture type**. Then the final channel value **final_value** is calculated by using the following formula:

```
multiplicator = 2 * ((console_value - dmx_full_range_min) /
(dmx_full_range_max - dmx_full_range_min))
final_value = auto_value * multiplicator
```

For example, consider a fixture with an **8-bit** IRIS channel (theoretical DMX values of **0-255**), but a limited dmx range defined in its fixture type (e.g. DMX values of **0-131**). With Auto 6M mode active…

- …sending an IRIS value of **0** (0 % of theoretical dmx range) from the lighting console leads to the final IRIS value being **0**, regardless of autofunctions or other factors (due to the multiplicator being **0** as well).

- …sending an IRIS value of **127** (50 % of theoretical dmx range) from the lighting console leads to the final IRIS value being **exactly the same** as **auto_value** (due to the multiplicator being **1**).

- …sending an IRIS value of **255** (100 % of theoretical dmx range) from the lighting console leads to the final IRIS value being **twice as big** as **auto_value** (due to the multiplicator being **2**).

Option 2: Property "Auto 6M Use Full Channel Range" is disabled:

Let *auto_value* be the value calculated based on autofunctions/smart zoom and *console_value* the one received from the console on the fixture's input universe. Furthermore, let *dmx_min* and *dmx_max* be the dmx limits as specified in the fixture's fixture type. Then the final channel value *final_value* is calculated by using the following formula:

```
multiplicator = 2 * ((console_value - dmx_min) / (dmx_max - dmx_min))
final_value = auto_value * multiplicator
```

For example, consider a fixture with an **8-bit** IRIS channel (theoretical DMX values of **0-255**), but a limited dmx range defined in its fixture type (e.g. DMX values of **0-131**). With Auto 6M mode active…

- …sending an IRIS value of **0** (0 % of limited dmx range) from the lighting console leads to the final DIMMER value being **0**, regardless of autofunctions or other factors (due to the multiplicator being **0** as well).

- …sending an IRIS value of **65** (50 % of limited dmx range) from the lighting console leads to the final IRIS value being **exactly the same** as *auto_value* (due to the multiplicator being **1**).

- …sending an IRIS value of **131** (100 % of limited dmx range) from the lighting console leads to the final IRIS value being **twice as big** as *auto_value* (due to the multiplicator being **2**).

- …sending an IRIS value of **255** (195 % of limited dmx range) from the lighting console leads to the final IRIS value being **almost four times as big** as *auto_value* (due to the multiplicator being **3.9**).