

## 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:

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

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 multiplier 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 multiplier 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 multiplier 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:

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

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 multiplier 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 multiplier 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 multiplier 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 multiplier being **3.9**).

From:

<https://manual.zacktrack.com/> - **zacktrack**

Permanent link:

[https://manual.zacktrack.com/doku.php?id=manual:auto\\_6m\\_use\\_full\\_console\\_range](https://manual.zacktrack.com/doku.php?id=manual:auto_6m_use_full_console_range)

Last update: **2023/03/27 15:51**

