

# Point data type

Datapoints are the most common types of data encountered when communicating with a BOS. 4 types of points can be listed.

- [List of different points](#)
  - [Input levels](#)
  - [Input strategy](#)
  - [Output definition](#)
  - [Additional notes](#)
- 

## List of different points

A Point is a [Data asset](#) with the tag **assetType="Data"** and the tag **dataType="Point"**

There is also a dedicated tag to define the type of data point: **controlType** with the following possible values:

- Numeric
- Boolean
- Enum
- String



**Numeric.** It is a number with a precision, it includes a unit and sometimes it contains a minimum and/or a maximum



**Boolean.** Its value can be "false" or "true". A tag is generally available to define a text for each of the states. *"On" & "Off, "Occupied" & "Unoccupied"*



**Enum.** It is an ordinal, i.e. an integer greater than or equal to zero, for which each value has a meaning. A tag is generally available to define a text for each ordinal. *"Up = 0", "Down = 1", "Stop = 2"*



**String.** The text can contain special characters. It can also be used to transmit complex data in JSON format. (A list of bookable rooms for example)

---

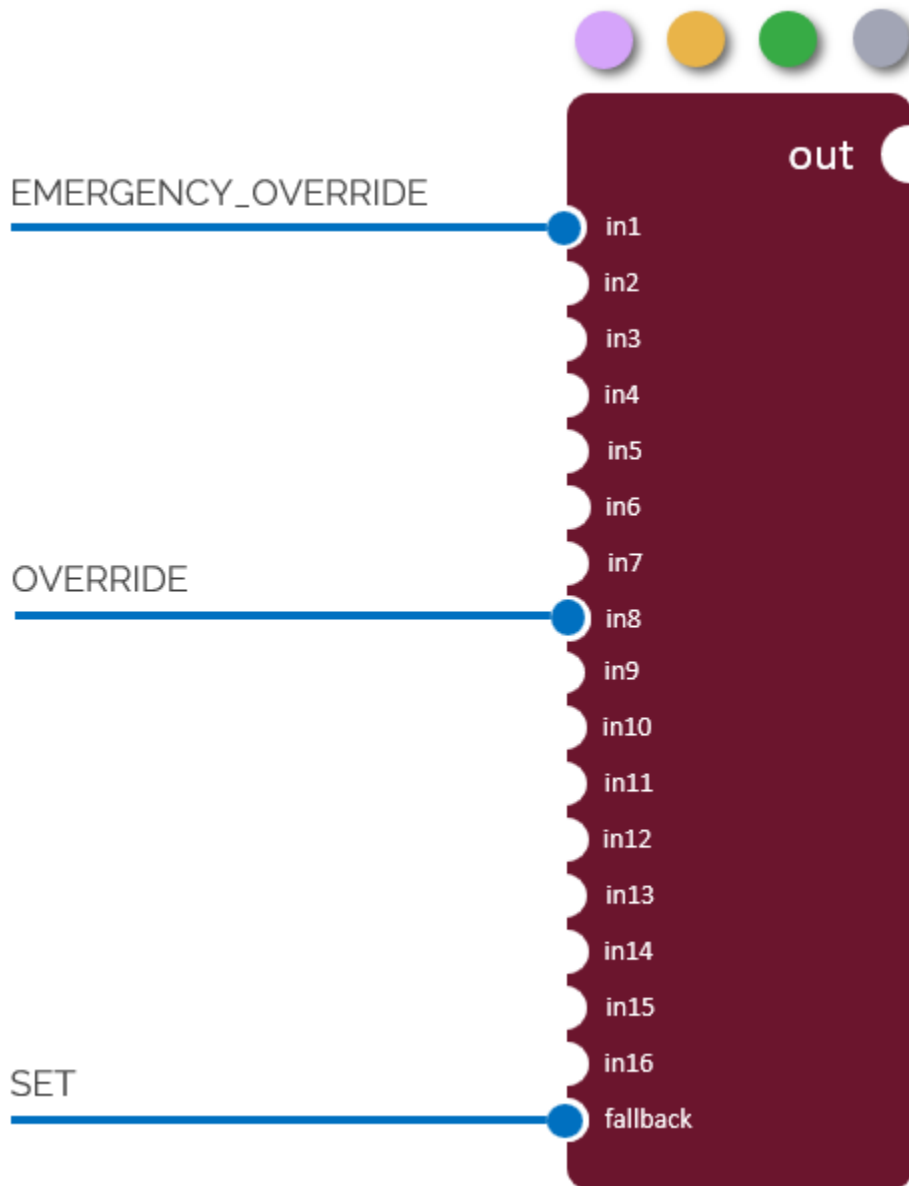
## Input levels

Each point can be **read-only** or **writable**. There is no dedicated tag for this information since a connector or the API can allow you to change the value of a point or not, it all depends on the configuration from the MSI, whether you're allowed to do it or not.

Usually writable points will match these conditions: `b:pointType='Command'` or `b:pointType='Setpoint'` or `b:pointType='Setpoint Offset'` or `b:pointType='Setting'`

A point has different levels of writing: 17 levels to be precise and only one single output. You will be able to write only at three different levels as shown below: in1, in8 and fallback.

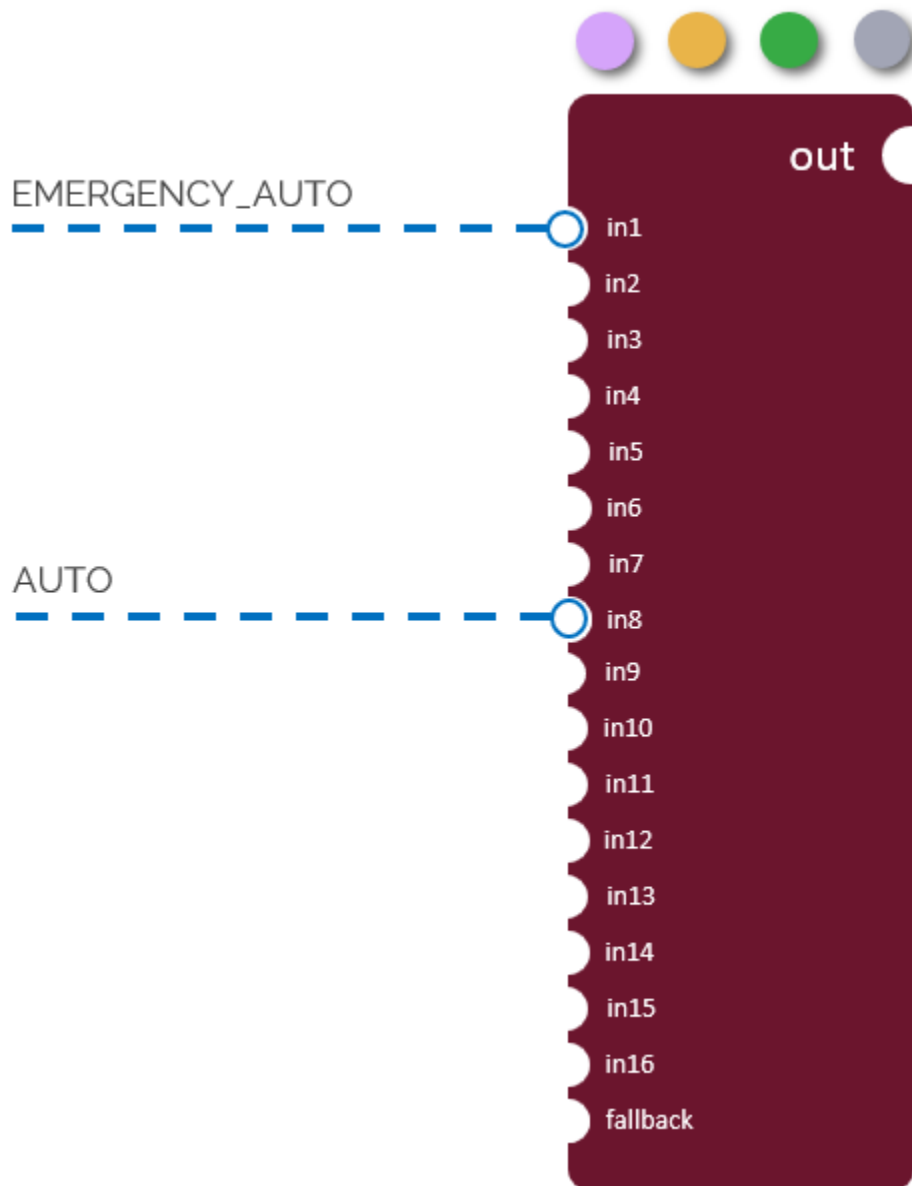
**The highest input level will win over the others.**



Important to know:

- The SET action is the only one persisted in the BOS, meaning that after a reboot, this value will remain. All the other levels will go back to default.
- There is no higher level than Emergency override

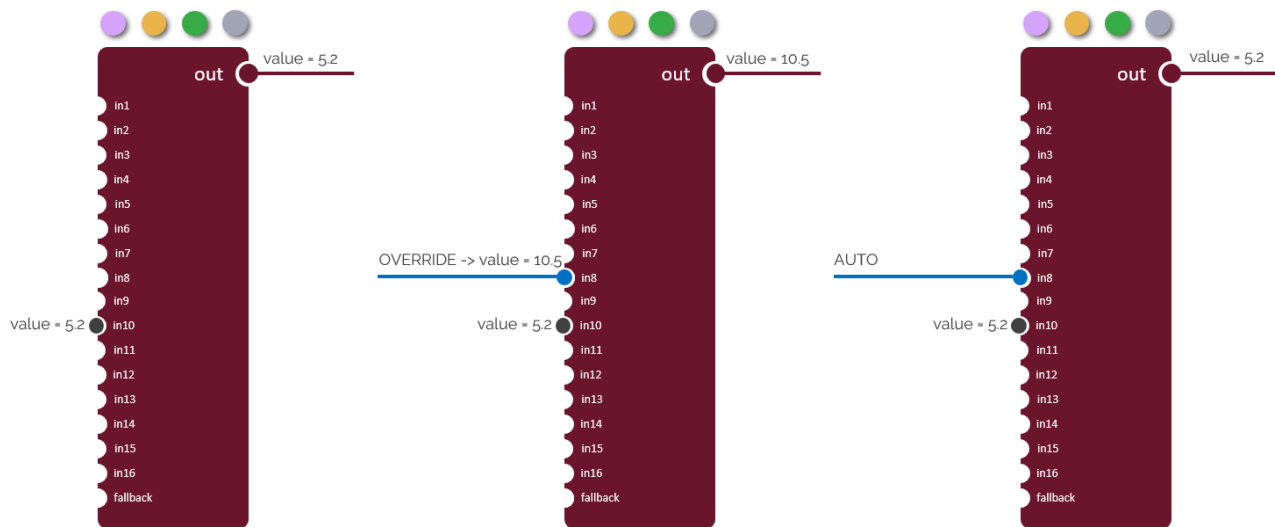
Both override has a release mechanism, to remove the override. In that case the closest lower level will take over.



Examples:

An Override action is triggered with the value 10.5. The level 8 is winning so the output takes the in8 value.

Then the Auto action is triggered, the in8 becomes inactive and the in10 is taking over.



## Input strategy

When you interact with the BOS, through the API or through a connector, you will need to choose between these three actions and their counter action.

You will need to understand how the local implementation is done and exchange with the local MSI as there are many possibilities.

The good practices (but that cannot be known for certain) are the following:

- Fallback is used when there is no specific logic before the point. It is used as a default value that you can change indefinitely calling the SET action.
- When there is an internal logic in the BOS prior to the point (or in the controller if it's Bacnet), the inputs are used from in9 to in16.
- You can override the logic for a certain duration. The logic can it self invoke the Auto action to release your override on certain event.
- The emergency override is let available to the local Facility Manager to override your own override. So use with caution the Emergency Override since there is no higher level. (The FM can still change the value by setting a new value with Emergency Override action from the supervisory interface)

## Output definition

The output of a point will contain three different parts:

- The value itself: a primary value like boolean, numeric, string or enumeration.
- The status. See [Linkspers statuses](#) for more information
- The active input

The active input will indicate which input is actively reported to the output.

- The format is as follow "@ <level>"
- Level range goes from 1 to 1, with 1 being the highest priority.
- Level 17 is identified by the key "def".

Linkspers will merge the status and the active input into the same field (through the API or the connector).

Different examples with exposed points on MongoDB:

```
status: "{ok} @ def"
timestamp: 2022-08-20T15:00:50.578+00:00
value: 0
```

```
status: "{ok} @ 10"  
timestamp: 2022-08-20T15:00:49.541+00:00  
value: "3"  
  
status: "{overridden} @ 1"  
timestamp: 2022-08-21T15:37:12.152+00:00  
value: "9"
```

---

## Additional notes

A point is a little more complex than the representation given above., there are additional features to calculate the output value from the inputs such as

- The presence of extensions that can include an algorithm and alter the active input value and the output status.
- The conversion applied to the read value

This is internal behavior to the BOS and do not alter a third party understanding of the exposed data.