

---

# **ppiSystemOverview**

*Release 1.0*

**Controls Group**

**Dec 12, 2024**



---

## Contents

---

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Device Scenes</b>	<b>7</b>
<b>3</b>	<b>Monitoring Configuration</b>	<b>11</b>
<b>4</b>	<b>Troubleshooting</b>	<b>15</b>
<b>5</b>	<b>Indices and tables</b>	<b>17</b>



Contents:



The `pplSystemOverview` (in the following referred to as ‘Overview’) is a user-oriented middlelayer device designed to provide an easy interface to monitor properties in a Karabo system; several types of properties can be checked against conditions set by the users.

In the following examples the Overview devices running in the LAS systems are used to show how the device works and how to easily configure it.

Properties can be logically grouped in three available device nodes, ‘inputs’, ‘internal’, and ‘outputs’. In each grouping node a property is described by a node which includes the property key name (‘Variable’), the property device (‘Device’), whether it is monitored (‘Is Monitored’) and whether its value is within specifications (‘Is Ok’ with key *isOk*), and its configuration parameters, see [Fig. 1.1](#).

In the device configuration editor the name of the property node key includes its type and an increasing number, e.g. ‘Analogue 00’, ‘Analogue\_abs\_range 00’, ‘Analogue\_abs\_range 01’, etc. Using the autogenerated ‘config’ scene, allows a more user-friendly way to access them, see [Fig. 1.2](#).

In case the expected conditions for a property are not fulfilled its Boolean property *isOk* is set to False, which in turns sets the corresponding grouping node *isOk* Boolean and the device Boolean *isOk* to False. This way, the property *isOk* of every device reflects the state of its monitored properties.

The main interface to the system is an upper-level Overview device (in the following referred to as ‘main device’) which monitors the *isOk* property of a series of Overview devices dedicated to monitor parts of the system. As described above, in case one property value in a monitored device is not fine, the subsystem device *isOk* is set to False, resulting in the main device to have also its property *isOk* set to False. For users thus it is sufficient to monitor only the *isOk* Boolean of the main Overview device to have a view of the state for the monitored properties in their system. A typical configuration is shown in [Fig. 1.3](#), where an Overview device can monitor both standard properties in Karabo devices and the property *isOk* of a sublevel Overview device.

In case in the main device the property *isOk* has False value, then a Zabbix alert can be sent to the users if the corresponding Boolean is set and if the Zabbix system has been properly configured to send notifications (previously the Nagios services were used instead of Zabbix).

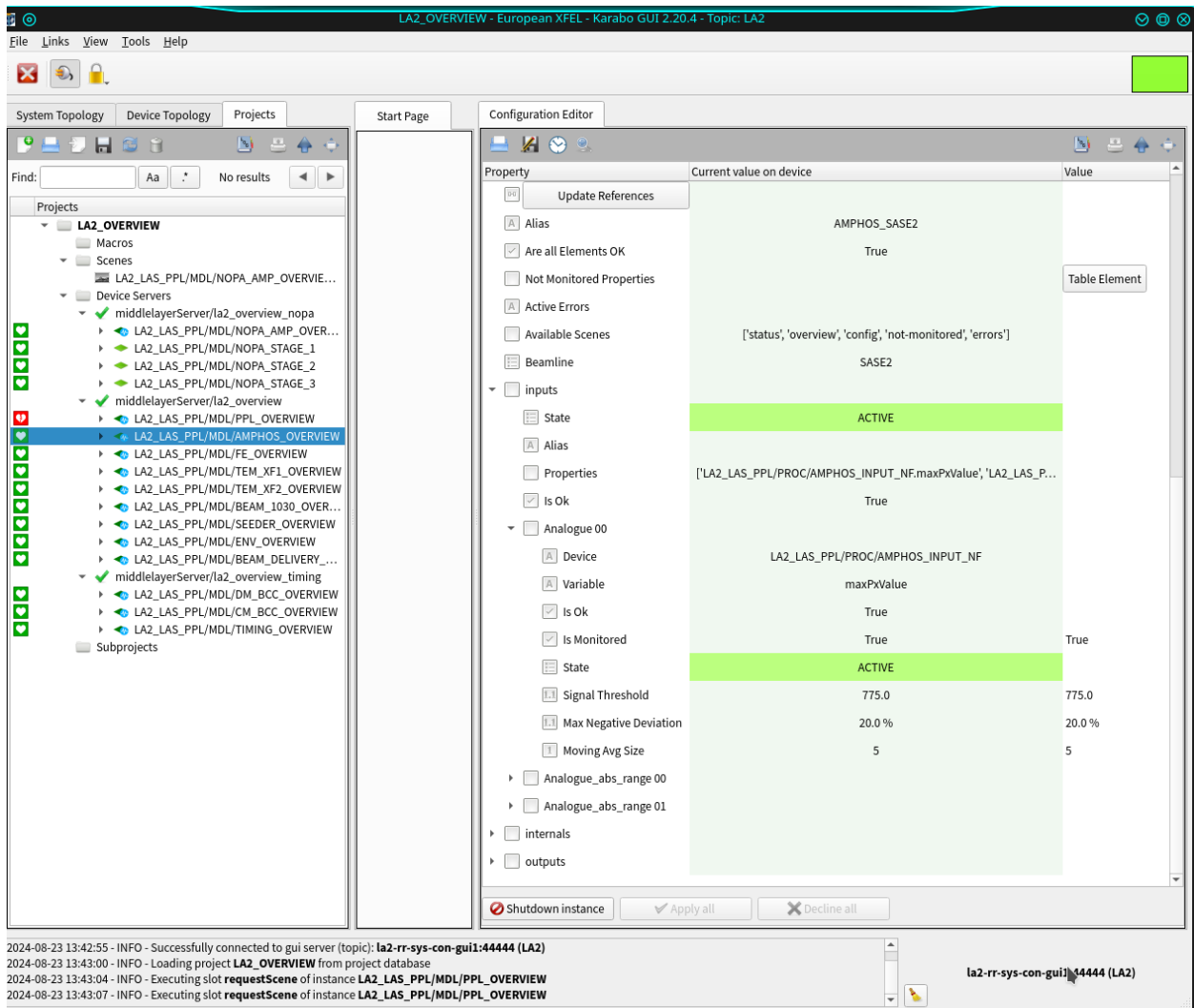


Fig. 1.1: The device nodes where the monitored properties are displayed.



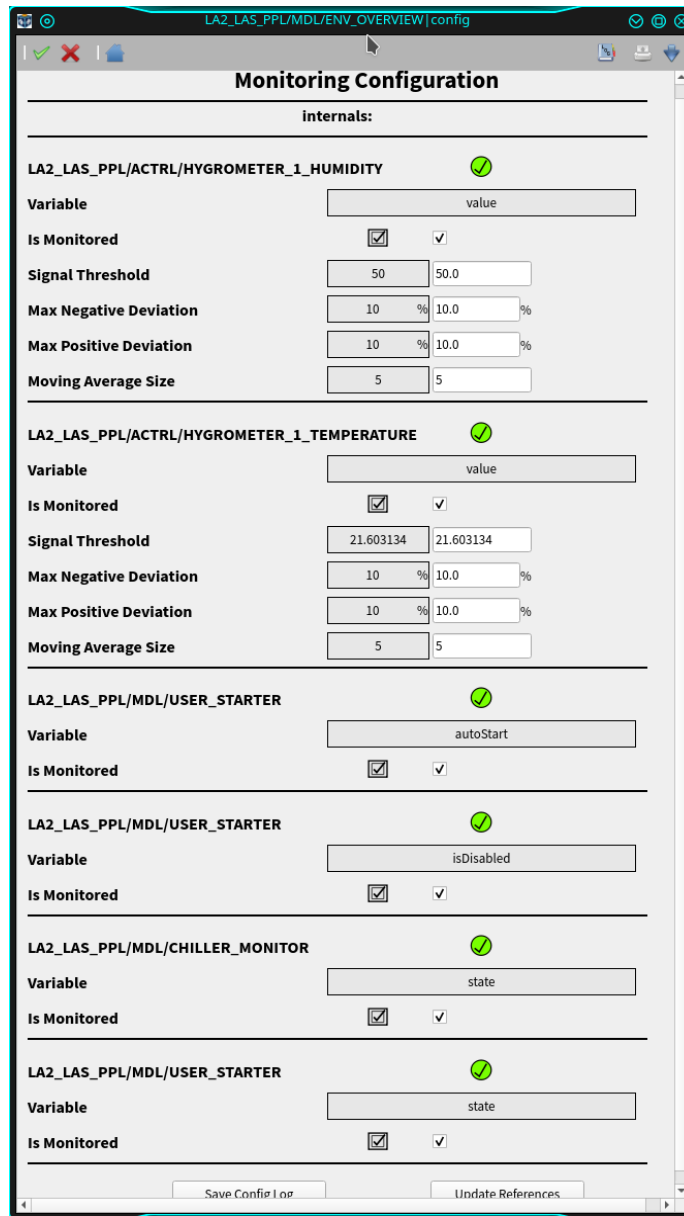


Fig. 1.2: The properties state and configuration can be accessed through the scene 'config'.

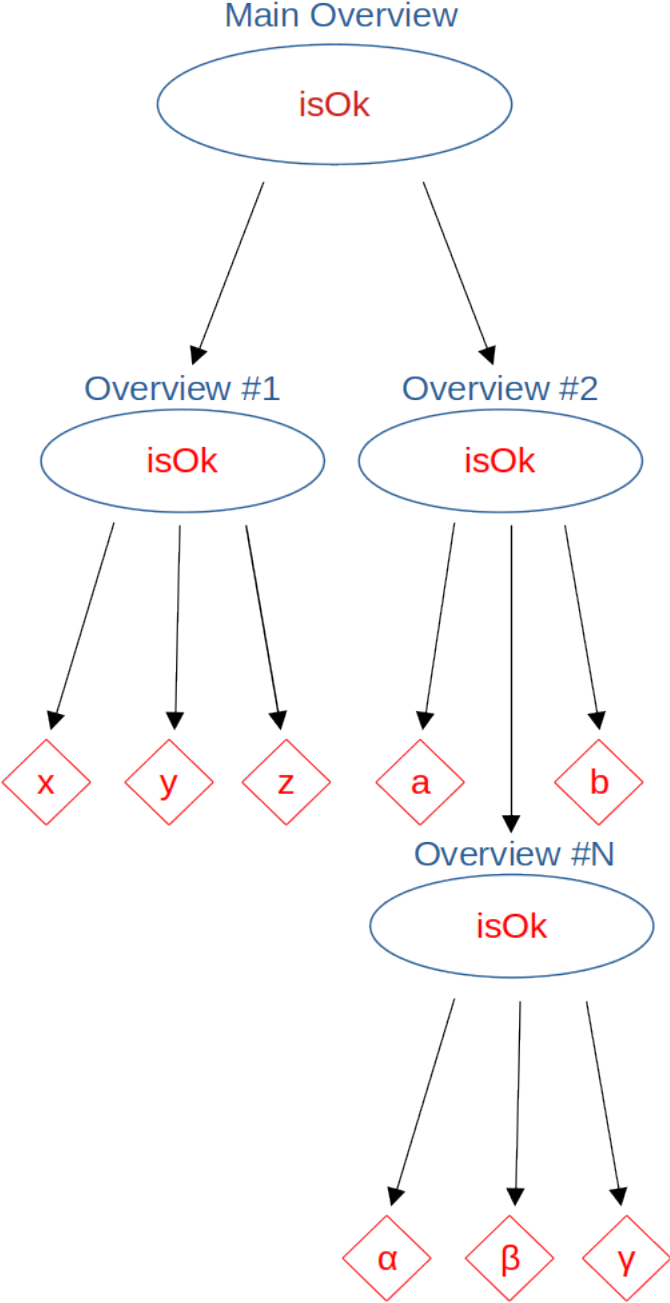


Fig. 1.3: Typical configuration of Overview devices.

## Device Scenes

Overview devices have five autogenerated base scenes which provide the full information of the monitored system. Double clicking on the device name in the GUI pops-up the default *status* scene, which provides a first view of the status of its monitored properties. Clicking on that scene pops-up the *overview* scene, see Fig. 2.1, which provides the state of the monitored devices and the link to the scenes displaying the properties in error (*Error Log*), the properties not monitored (*Not Monitored Properties*), and to configure the monitoring (*Monitoring Config*), see Fig. 2.2:

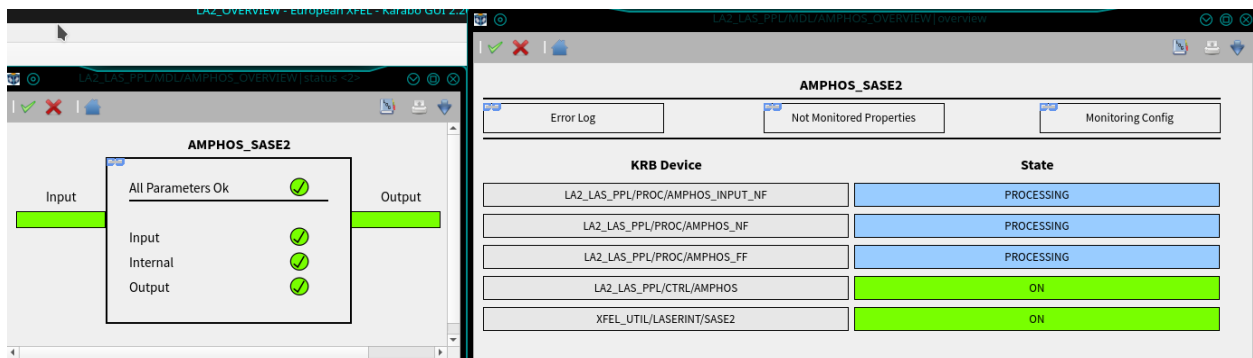


Fig. 2.1: Default *status* and *overview* autogenerated device scenes.

For an optimized layout, visualization base scenes can be overwritten in specific derived devices, as in the case of the main overview scene for LAS systems. An example for the LA2 system is shown in Fig. 2.3; a similar layout concept can be used in other Karabo systems. In this example, each block represents the status of a monitored Overview subsystem; clicking on its scene-link will open its default scene. Also shown in the block are the *isOk* status of the corresponding Overview device and the Boolean to enable its monitoring.

The relevant properties to monitor in an Overview device should be added in its hardcoded configuration; users can anytime disable the monitoring of a property which was initially included in the configuration. The key *unmonitored* (labeled as *Not Monitored Properties*) reports those properties; in case the *isOk* state of an Overview subsystem is included in the monitoring, then that key reports also the properties whose monitoring was disabled in that monitored Overview device. This means that the *unmonitored* key in the main device shows the properties not monitored in the entire system, see Fig. 2.4. Each entry of the scene table reports the key of a non-monitored property, the ID of the device device the property belongs to, and the ID of the Overview device in charge of its monitoring.

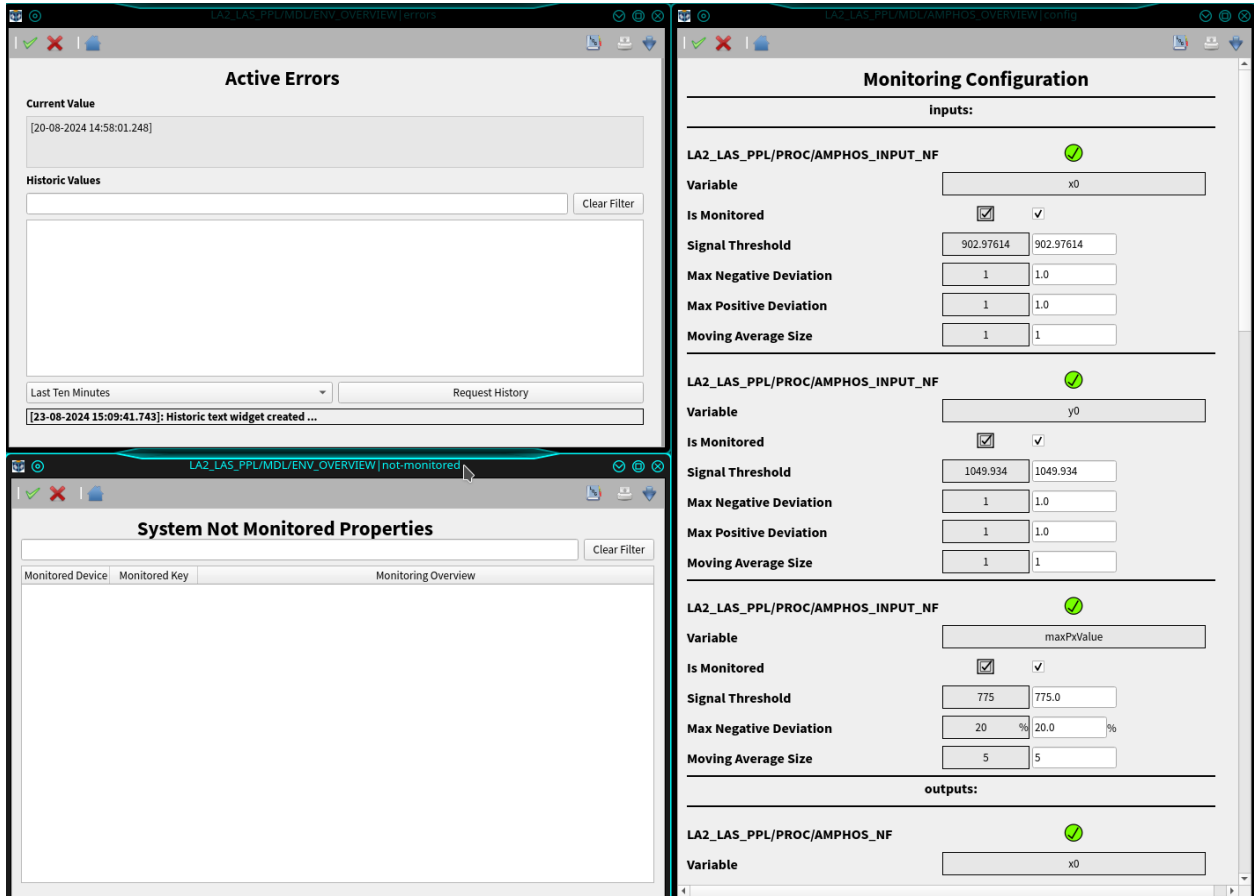


Fig. 2.2: Scenes for displaying the properties in error (top-left panel), the not monitored properties (bottom-left panel), and to configure the monitoring (right-panel).

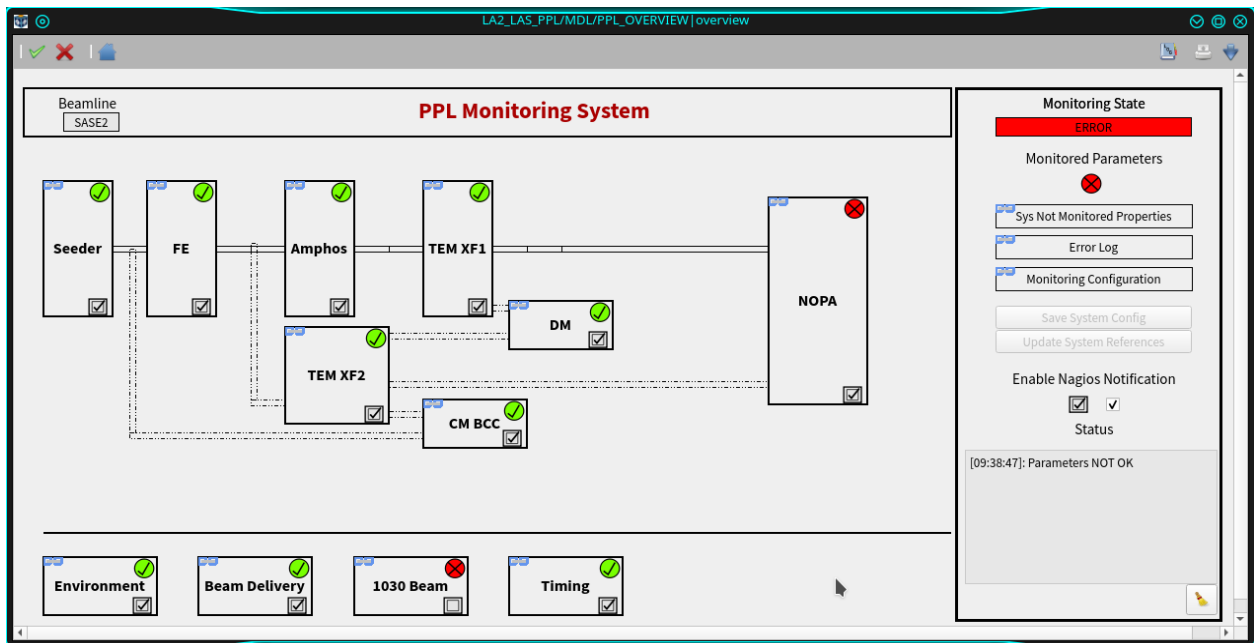


Fig. 2.3: Dedicated scene overwriting the base *overview* in the LAS LA2 system.

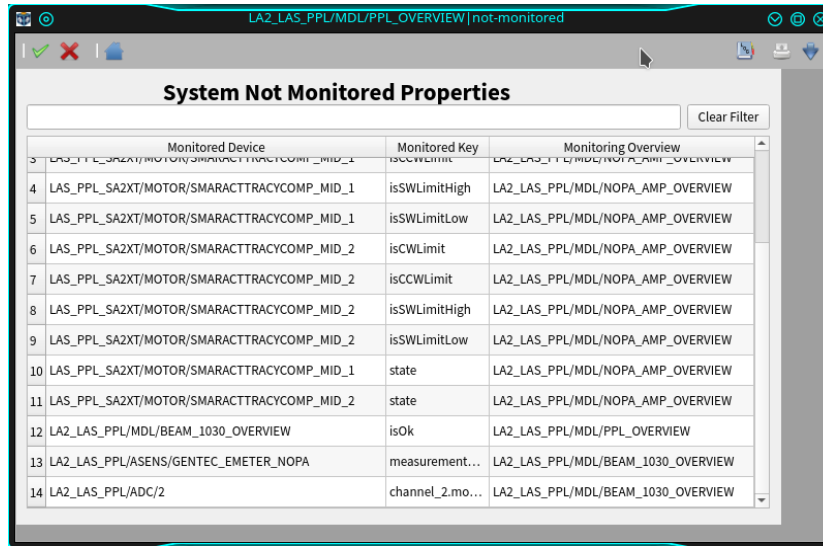


Fig. 2.4: Scene to show which properties are excluded by the users from the monitoring.

The scene to tune the configuration of the properties included for monitoring is described in details in the next section.

An example of scene showing the properties which are not fulfilling the configured conditions is displayed in Fig. 2.5. Every time a monitored property goes to error or is recovered from a previous error, the property string *activeErrors* reporting all properties currently in errors is updated and presented with a new timestamp. Being this information contained in a Karabo string allows the users to visualize the property history up to the last week; The investigation for values older than one week can be done via the Grafana interface to the device property database.

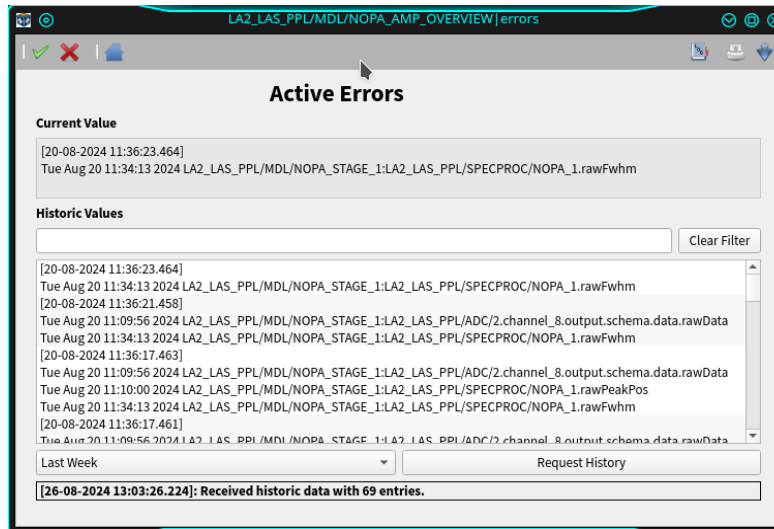


Fig. 2.5: Scene to show the properties currently not fulfilling the configured conditions.



---

## Monitoring Configuration

---

The Overview device has an autogenerated scene for tuning the monitoring configuration. An example is shown in Fig. 3.1. The monitored variables are grouped according to the device node they belong to (i.e, *Inputs*, *Internals*, *Outputs*).

To include a variable to a monitoring Overview it should be added (by CTRLS) to the hardcoded configuration of that device, along with the proper monitoring type class for that. The monitoring classes currently available in the overview device are described here below together with their available settings:

- *AnalogueElement*:  
monitors a float to be above a *Signal Threshold* value; tolerance to this condition is provided by allowing a deviation down to a an absolute value below (*Max Negative Deviation*) the threshold. The deviation is given in as percent value of the threshold,
- *AdcElement*:  
similar to the *AnalogueElement* case, except that it monitors the maximum signal of an array,
- *AnalogueAbsRangeElement*:  
monitors a float to be above a *Signal Threshold* value; a tolerance to this condition is provided by allowing deviations up to an absolute value above (*Max Positive Deviation*) and below (*Max Negative Deviation*) the threshold,
- *AnalogueRangeElement*:  
similar to the *AnalogueAbsRangeElement* case, except that the deviation range is given as percent of the threshold value,
- *UIntegerElement*, *DoubleElement*, *StringElement*:  
monitors an unsigned 64-bits integer, double or string to have a configurable *Reference* value,
- *UIntegerLimitsElement*:  
monitors an unsigned 64-bits integer to be within configurable “Min Value” and “Max Value” values,
- *BooleanElement* (*FalseBooleanElement*):  
monitors a device variable which is expected to be True (False),

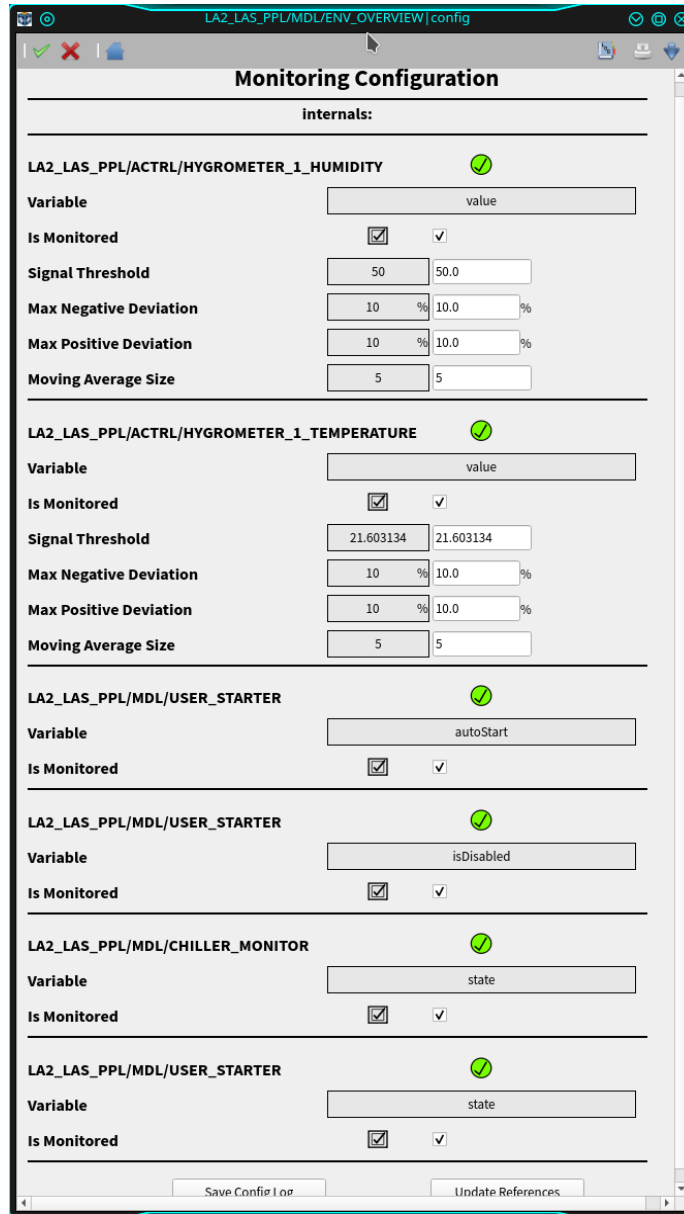


Fig. 3.1: Scene to show which properties are disabled from the monitoring by the users.



- *StateElement*:

monitors a device state which is expected to be in a typical running condition (at the moment, device class dependent).

Analogue values are typically fluctuating due to different kind of conditions, such as due to the environment (e.g. thermal noise in the acquisition line) or to the physical process generating the signal (as for the amplified pulse-probe laser). To avoid fluctuations giving rise to warnings, moving averages are considered for that kind of variables. The size of the those average samples *Moving Average Size* is tunable by the users up to 20. Setting the size to one disables the running average feature.

It is the responsibility of the user to choose the proper monitoring class for a specific variable, and to tune its corresponding settings.

In the configuration scene, for each variable its quality state Boolean *isOk* is displayed close to the corresponding device Id, as well as its key name in the monitored device. By default its monitoring is enabled, but the user has the possibility directly from the scene (or from the device configuration editor) to disable it by unsetting the corresponding *Is Monitored* Boolean (*isMonitored* key). Every change of this Boolean results in the immediate update of the list of not monitored properties. Disabling the monitoring of a property turns immediately its *isOk* state to True.

A change of the Boolean *isOK* triggers the evaluation of the grouping node state it belongs to; the node *isOK* state will be False if at least one monitored variable in the node is in a bad state. A change of the node *isOK* value triggers the evaluation of the device *isOk* Boolean, as described in Fig. 3.2:

At the bottom of the scenes two slots (buttons) are present to help configuring the monitoring:

- *Update References*

The references are updated reading from the running system the current value of the monitored properties; e.g., when monitoring a temperature sensor (via the class *AnalogueRangeElement*) the current reference value will be set as the new reference,

- *Save Config Log*

The monitoring configuration is saved online, and whenever the Overview device restarts that configuration will be set back to it.

In the device configuration editor of the main Overview device (and also directly in the LAS main overview scene) the extra slots *Save System Config* and *Update System References* save the online configuration and retrieve the references recursively through the entire system, respectively.

In the LAS environment the so-called *slave* experiment in the two-laser-beamlines systems is the experiment which does not receive the pulse-probe laser synchronized to work with the XFEL x-rays; when this is the case some specific devices will not operate within the nominal running parameter values (e.g. having only background signal in the diagnostic cameras), and thus the corresponding Overview devices will have the state *isOk* unset. In this situation those devices have to be disabled (enabled) in the monitoring when the instrument switch from *master* to *slave* (and vice-versa) operational mode is done (typically in the morning and the evening). This is implemented automatically by the Overview device after inserting (by CTRL-S) those devices in a special list in the Overview configuration.

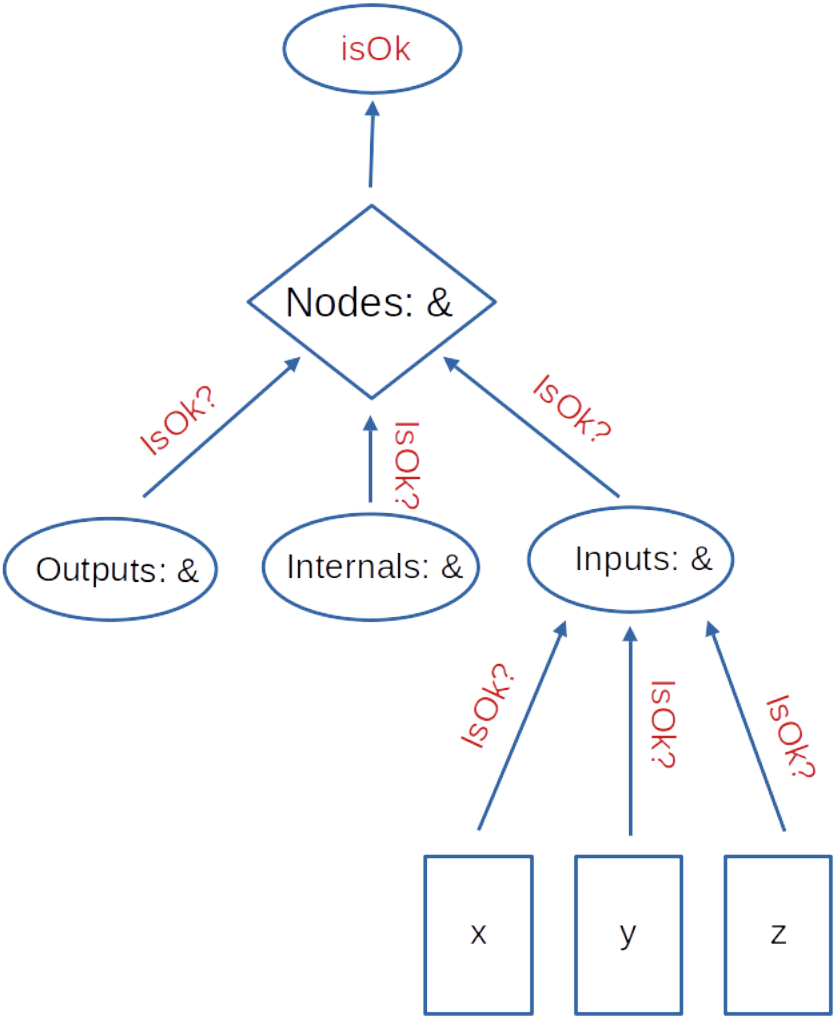


Fig. 3.2: Evaluation of the Overview device monitoring Boolean *isOK* state.

---

## Troubleshooting

---

Some typical situations were observed to trigger a False property *isOK* state, and could not be easily fixed from hardware side, e.g. by increasing the exposure time of a diagnostics camera:

- Analogues typically fluctuate, and their value can overshoot and/or undershoot the region of accepted values.

**Possible solution:** Enlarge the region of accepted values if possible. Also, increasing size of the running average sample should reduce the sensitivity to outliers.

- During a specific experiment a monitored parameter varies sizable due to modifications of the running condition of an experiment, e.g. changing periodically the number of PPL pulses while monitoring the intensity of the laser beam. Ideally using a normalized signal would be ideal to handle this kind of situations, but unfortunately a Karabo device with this functionality is not present at the moment.

**Possible solution:** Disable the monitoring of that property during those specific running conditions.

- After preparing the setup for a new experiment the Overview devices start showing the monitored properties in bad state. This could be due to different running conditions in the experimental area, resulting in the monitored parameters being outside of the expected range.

**Possible solution:** Reload new reference values from the system by clicking the button “Update References”.

- During a new experiment some properties, or even an entire Overview device (checking a specific section of the experimental area, no more in use) should not be monitored, but they will be need in the future. A similar situation can also appear when the connection of Karabo to a section of experimental area is broken, due e.g. to hardware or infrastructure failures.

**Possible solution:** Those properties (or Overview device) can be easily disabled from the monitoring, letting the Overview monitoring the system while disregarding those properties. Saving online the new system configuration allows to recover the new setting in case of restart of devices.

- If a device, working during commissioning, hangs in the initialization state when restarted, then some needed connections to devices (to monitor the properties) cannot be established.

**Possible solution:** Check that the needed devices are actually online.



## CHAPTER 5

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`