# Alarm generation conditions

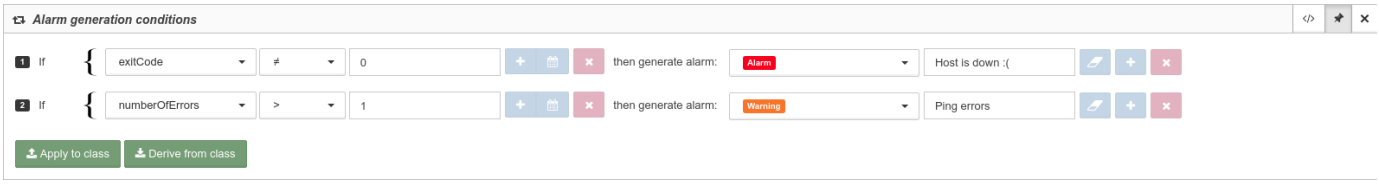
**Incident functionality is disabled by default. To anable it please add or modify the following parameters:**

- **"conditional_incidents_enabled" : true** in configuration file **/etc/saymon/saymon-server.conf**;
- **enableConditionalIncidents : true** in **/usr/local/saymon/client/client-config.js** ( **/usr/local/saymon/js/client-config.js** in some installations).

Both **Alarm generation conditions** and **State change conditions** can generate incidents.

While these features look very similar:

? Unknown Attachment




... they have got some core difference we will try to describe below.

## 1. Priority of conditions

By default (if **Alarm generation conditions** are not defined) the incidents are generated for the objects in `ALARM`, `OBJECT IS OVERLOADED` or `NO DATA ON OBJECT` states, which are defined in **State change conditions**.

The presence of **Alarm generation conditions** disables generation of incidents by **State change conditions**, but does not disable states changing rules defined.

## 2. One incident or several?

An object can be in one state at a time (if it's not Schrцdinger's cat of course), that is why:

**State change conditions** create only one incident for one object at a time.

**Alarm generation conditions** allow to create several incidents for one object.

For example, there is the object to check temperature (T) and humidity (H).

You set the following **State change conditions**:

- if `T > 30`, then `ALARM`;
- if `H > 50`, then `OVERLOAD`;
- else `WORKING`.

The object gets `T = 31, H = 51` values.

In this case:

- the object will be transferred to ALARM state;
- CRITICAL incident will be generated for the ALARM state.

Now you add the following **Alarm generation conditions**:

- if `T > 30`, then `CRITICAL`;
- if `H > 50`, then `WARNING`.

The object gets `T = 31, H = 51` once again.

In this case:

- the object will be transferred to ALARM state;
- there we will be created two incidents: CRITICAL and WARNING.

When generating alarms, four default levels are available, as well as manually created levels:

- ALARM
- CLEARED
- MAJOR
- WARNING

## 3. Incidents autoclearing and clear conditions

By default (if **Alarm generation conditions** are not defined) the incidents are generated for the objects in ALARM, OBJECT IS OVERLOADED or NO DATA ON OBJECT states, which are defined in **State change conditions**.

In this case the incident will be cleared if the object changes its state to any other state.

The presence of **Alarm generation conditions** disables generation of incidents by **State change conditions**.

In this case the incident will be clearedif:

- the clause to generate this incident is not valid anymore

or

- this incident has got its own clear condition added with  button and this condition is valid.

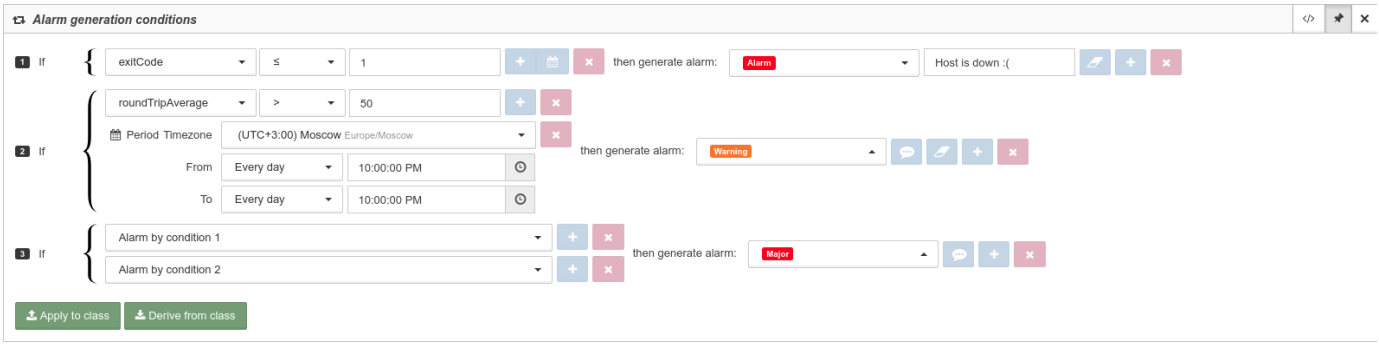
## 4. Actions on state changes

Actions on state changesfunctionality depends on **State change conditions** only.

Presence or absence of **Alarm generation conditions** do not affect it.

# Synthetic alarms

**Synthetic alarm**- an alarm generated by one or several regular alarms embedded in it.




When generating synthetic alarms, three states are possible:

- CRITICAL
- WARNING
- MAJOR

If several synthetic alarms are specified for one object, then the lower it is in the list of alarms, the more priority it has.

There is a list of rules regarding the mechanism of correlation of synthetic alarms:

1. One alarm is specified for each object.

2. After receiving the clearing message, the alarm is colored green with the status"clear" and:
   -if a similar alarm occurs before n-minutes (n - configurable trashhold time), then the alarm resumes;
   -if a similar alarm occurs after n - minutes, then a new alarm is created.

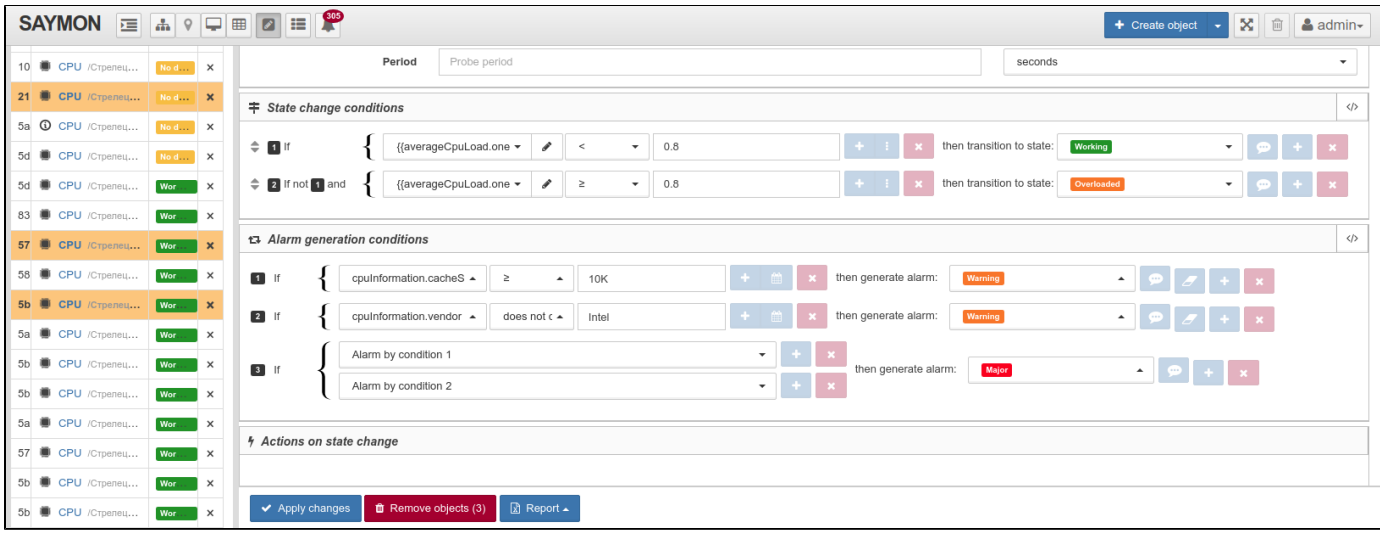
3. The transition from the list of active to the list of historical alarms occurs in m-minutes after receiving a clearing message (m - configurable time during which the alarm is in the list of active alarms after its completion).

4. The time of displaying the completed alarm in the active list should be more than or equal to the time of trashhold:**m n.**

5. If a script is configured for two objects and:
   -an alarm occurs on both, then a synthetic alarm and individual nested alarms for each object aregenerated;
   -an alarm occurs only on one object, then a synthetic alarm, nested alarms for each object and individual alarm for each object are generated.

6. A synthetic alarm closes after closing of all nested alarms.

If it is necessary to generate the same synthetic alarm for several objects, use the "Search and bulk operations" window.




Synthetic alarm can be formed on the basic of two or more synthetic alarms, generating a double level of nesting. With the example of motion sensors, this would look like this:

1. all sencors were triggered;
   1.1. penetration;
      1.1.1. the door 1 is open;
      1.1.2. motion detected;
   1.2. hacking device;
      1.2.1. the door of device is open;
      1.2.2. external power is off.

To close Priority 0, it is enough to make a correlation for the door sensors and motion sensors.