# Alarm Line

After chapter 3.1.5.2 Use Case: Alarm control (old alarm line explanation)

### Unified Controls

#### Use Case: Alarm Line custom solution

Description

For accessing alarms in general an alarm view is provided. As mentioned before (3.1.5.2), when it needs to be displayed in a static screen window, i.e. the header, the simplified appearance style is a good solution. But this is not recommended if the alarms need to be displayed in a screen that is getting reloaded often.

Solution

Using an Alarm Line to display only the latest triggered alarms (i.e. only 3 out of 8.000 pending alarms) can be a good solution for avoiding using an alarm control when you do not need all its functionalities. This is an open-source scripting solution provided free of charge via **Github** and is a good option if you only need to display the alarm name, raise time, status and address of the latest alarms. You can implement the Alarm Line custom solution with the following steps:

1. Create a new Global Module in your TIA Portal project and copy the code from **Alarmline\_GlobalDefinition.js** to the Global definition area. Here, the Alarm Manager class is defined, and it has several methods and properties (initializes the instance of the Alarm Manager with the given options, starts the alarm subscription, sets the sorting order and stops the subscription)
2. Add the new function “UpdateActiveAlarms” and copy the corresponding code from **Alarmline\_UpdateActiveAlarms.js**. This function will update the alarm tags based on the provided array of alarms. It creates a tag set and updates the values based on the alarms.



1. Create a new Scheduled task, i.e. “AlarmUpdate”, and select the tag “@SystemActivationState” as trigger. This task will update the alarms, it starts the alarm subscription based on the system activation state.



1. In the Events > Update of the scheduled task, copy the **ScheduledTask\_GlobalDefinition.js** code to the global definition and **ScheduledTask.js** to the event itself
2. For the visualization of the alarms, you can configure it with basic objects and elements from the toolbox (see Demoproject)
3. Create the required HMI tags for displaying the Alarm Line values and dynamizing properties. The number of tags needed will vary depending on the maximum number of alarms you want to display at the same time.



With this approach, there are some parameters that can be modified to better suit your requirements:

* **Max. number of alarms**: the maximum number of alarms displayed at once can be modifed in the Global definition of the Alarmline global module, in line 164. Take into consideration that, if more alarms are to be displayed, more HMI tags and basic elements are needed for displaying them.



* **Alarm language**: in the scheduled task event code, the language used for the alarms can be set by changing the value (decimal language ID) in line 7. In this example, “1033“ refers to English (here you can find [further language IDs](https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/available-language-packs-for-windows?view=windows-11)). You can also use the value “127”, which is the default languege configured in your HMI.



* **Alarm filter** : in the global definition of the scheduled task,the filter for the alarms is configured. This can also be modified so that it fits your needs.
* **Update interval** (optional): in the Scheduled task Event code, you can add the parameter “delayInMilliseconds” to the Alarm Manager() call. The suggested time is 250ms or more, since below this value there is no real benefit. Notice that, when setting it to a higher number, it does not mean that you will not be updated in time. This paramater works as follows: imagine that we have set 10.000ms as delay, whenever there is a new alarm raised it will be sent directly, as soon as possible, to the visualization. If after 2s another alarm is coming, it will not be sent to the visualization directly, but it will be visible after 8s (10s delay – 2s = 8s). This will allow us to see the alarms in time and also prevent overflowing of the system. This is an optional parameter, so you can choose the delay you prefer or not use it at all.

