Alarm Limits by Shift Example

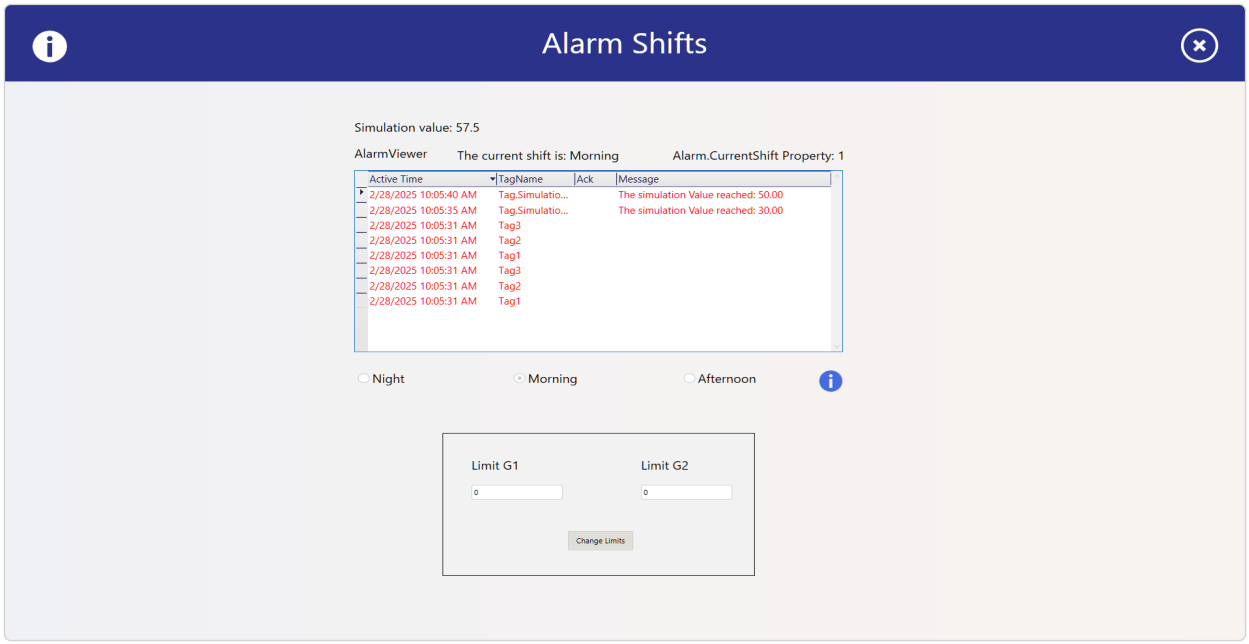
Apply alarm limits by work shift.

Download the solution Alarm Limits by Shift Example.dbsln.

This solution example demonstrates how to configure alarm limits.

Summary

We will introduce two distinct configurations for alarm limits. The first pertains to the setup of shifts, while the second addresses the management of AlarmItems grouped by shared limits, offering a method that can be used to easily adjust the limit for hundreds of tags.

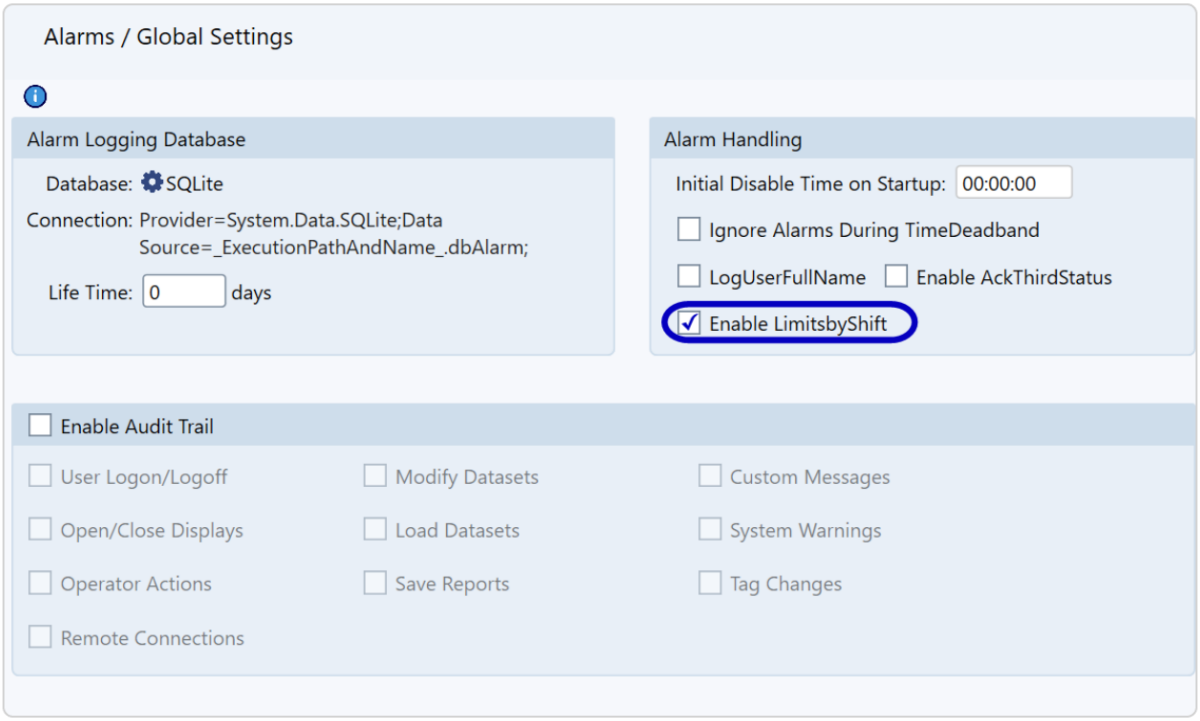


Limits by Shift

We can preconfigure up to three limits for a single AlarmItem, but only one of them will be effectively utilized at a given time, determined by the shift configured in the Alarm.CurrentShift property, as outlined in the following table.

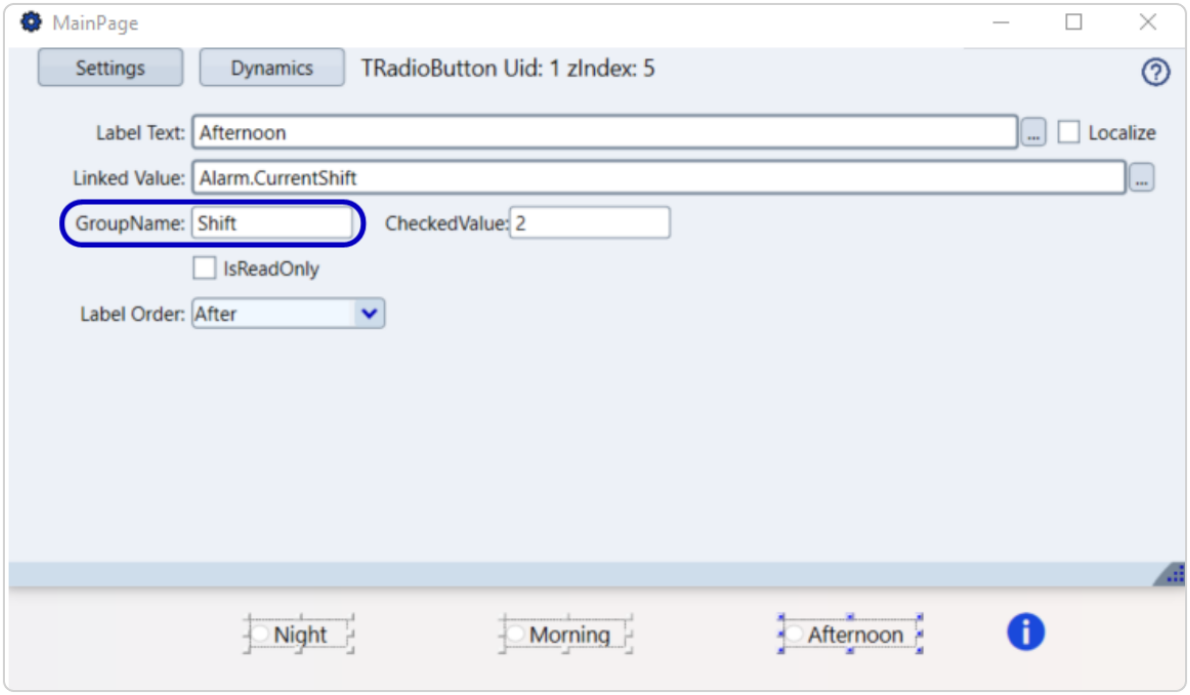
Limit Name	Value
Limit	3
Limit1	1
Limit2	2

It's important remember that for this functionality to work, you need to select the checkbox at **Alarms** **Global Settings** **Alarm Handling** **Enable Limits by Shift.**

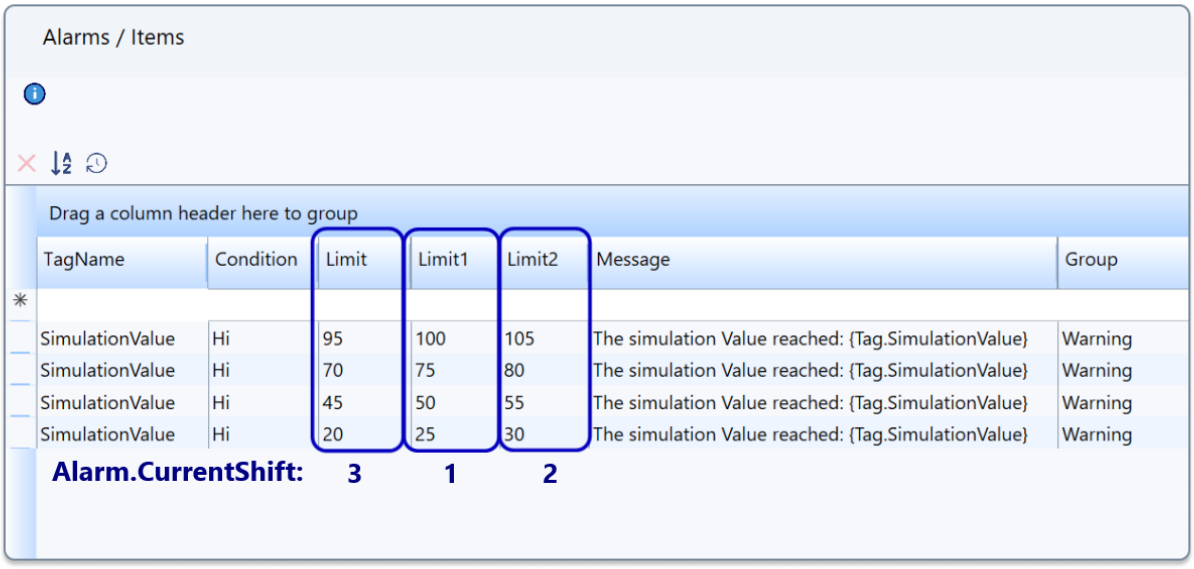


Configuration Example - Limits by Shift

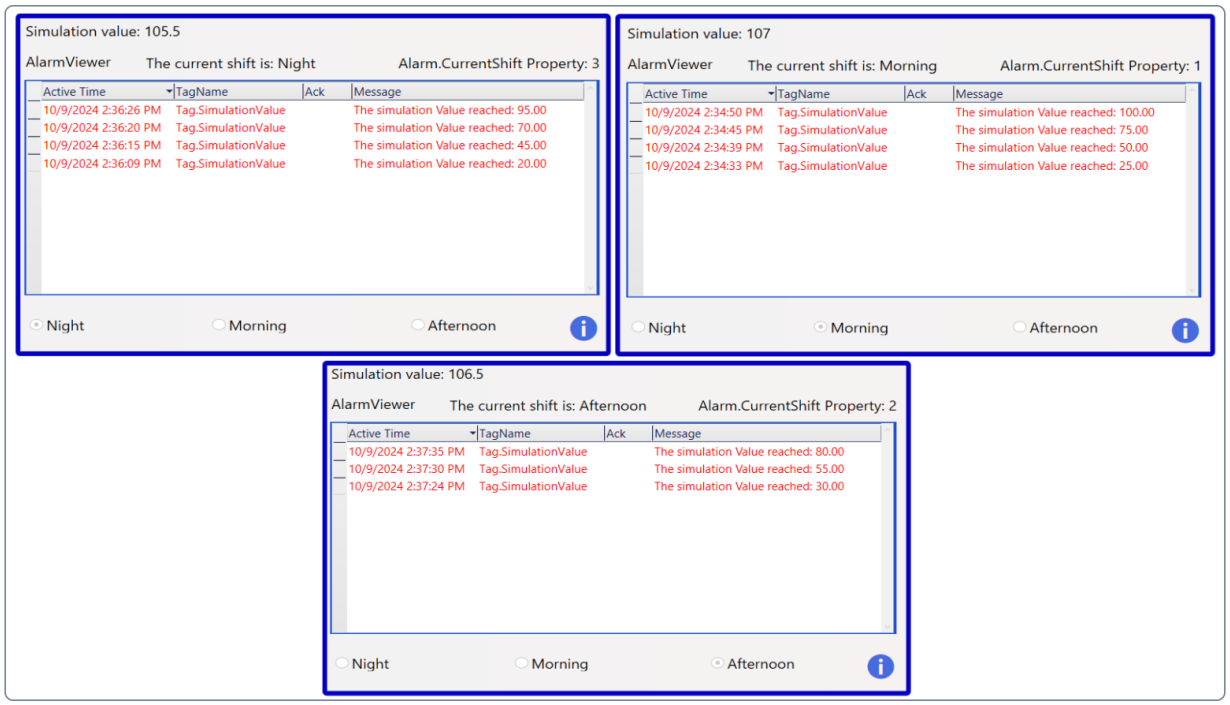
To switch between shifts, this example uses three RadioButton, one to each shift and all in the same group. When one is selected, the property Alarm. CurrentShift will change to respective value, and the new configured Limit will set. To do it, after adding the RadioButton controls to display, double clic k them and fill the field **GroupName** with the same name to all of them, defined as "Shift" in this example. The follow image illustrates this config.



Next, at **Alarms Items**, the Limits configuration image illustrates the predefined alarm limits linked to the Alarm.CurrentShift property. For each shift, a different limit is applied: 3, 1, or 2. These limits are assigned based on the shift's respective value.



Finally, the runtime results image displays the alarm limits behavior changing as the shift is selected. The active alarms reflect the specific limits set for each shift as per the Alarm.CurrentShift property.

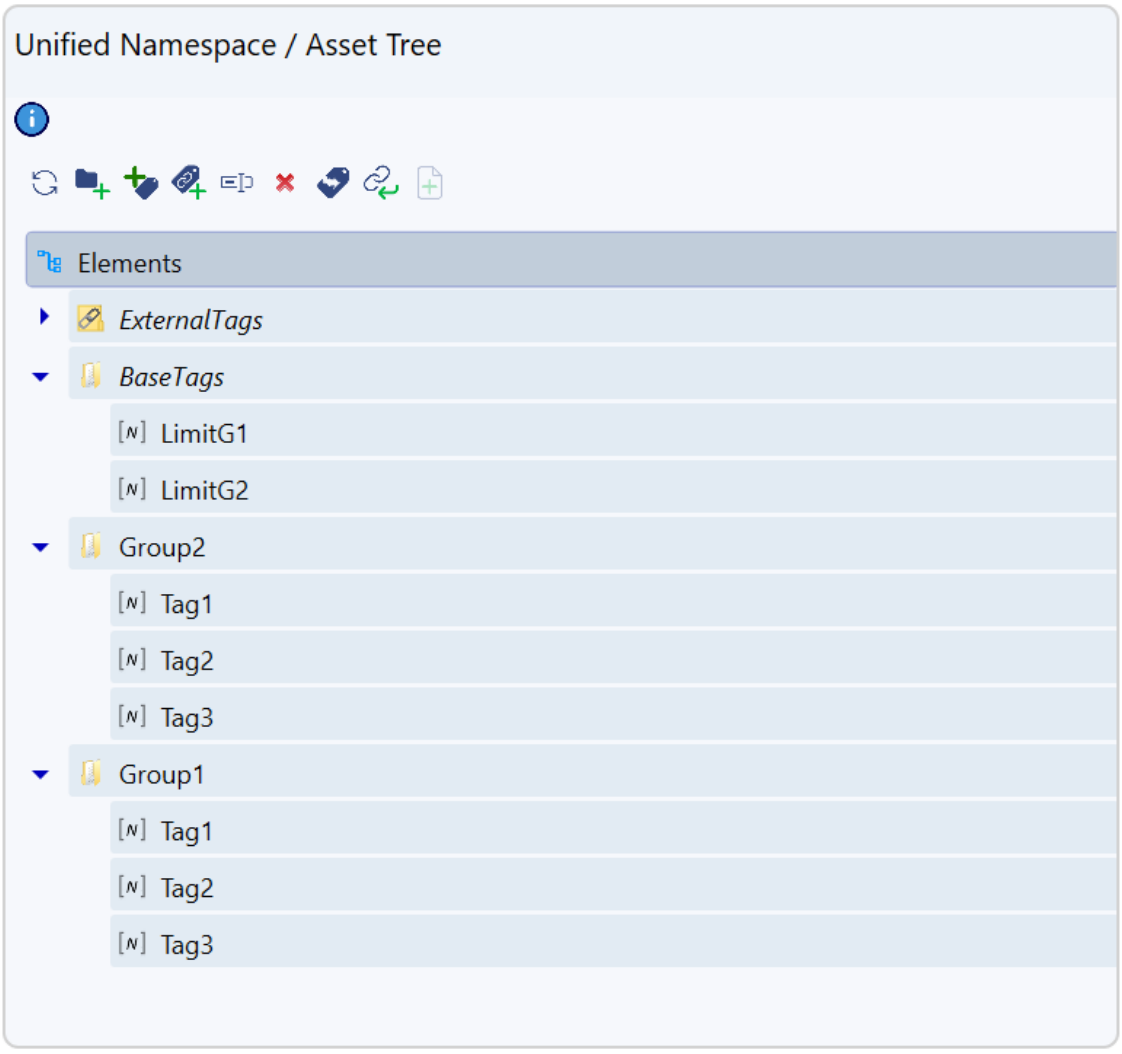


Shared Limits

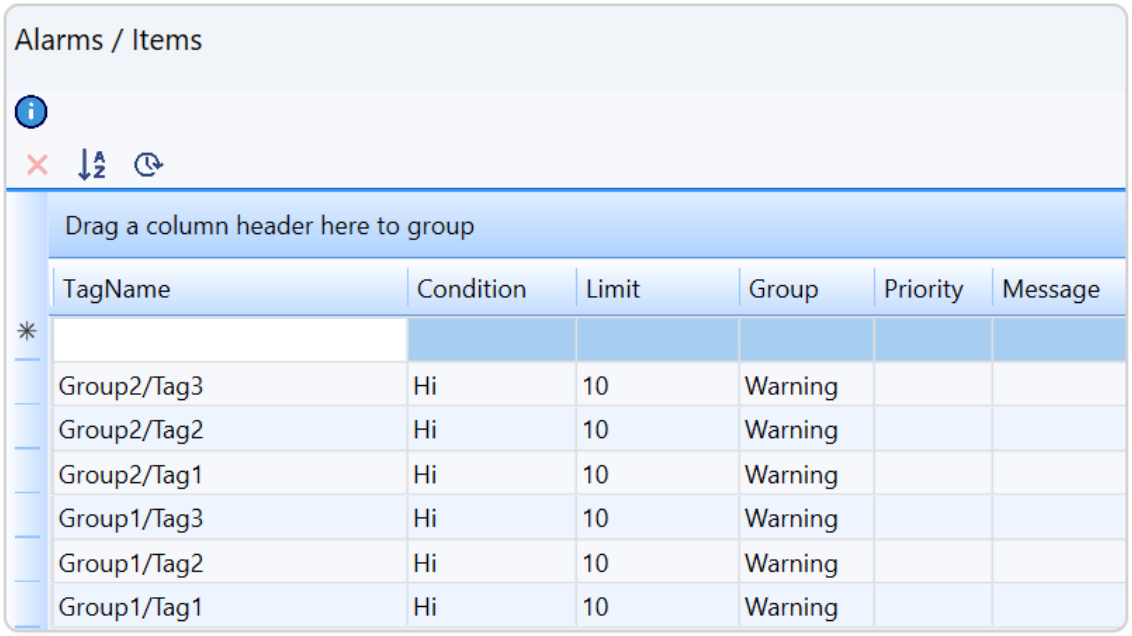
In some scenarios, it may be necessary to establish an alarm limit linked to a specific tag, especially when dealing with a group of tags that have identical alarm limits. That way, when a change is needed, all tags can be adjusted simultaneously. While it is feasible to alter the limits individually at runtime for each tag, a more effective method would be to link a group of tags with similar characteristics to each other. In this sample document, we will delve into how you can alter the alarm limit for a group of tags at runtime.

Configuration Example - Shared Limits

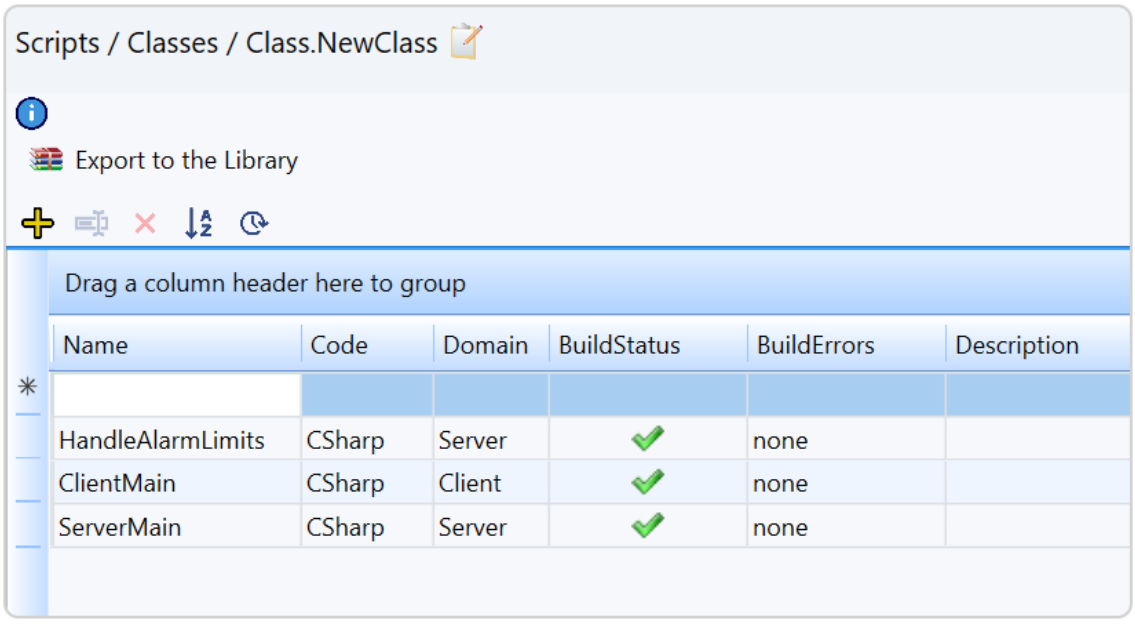
Firstly, we created two assets called Group1 and Group2, each one with three tags. We also created two tags to set the limits for each group, as shown in image below:



Items were created to each tag with "Hi" condition, and the initial limit was defined as 10:



After this, in Scripts Class, a class called "HandleAlarmLimits" was created, like the following image:



Inside the previously created class, the method "changeGroupsLimits" was created, where the groups limits were linked with each tag of the respective groups:

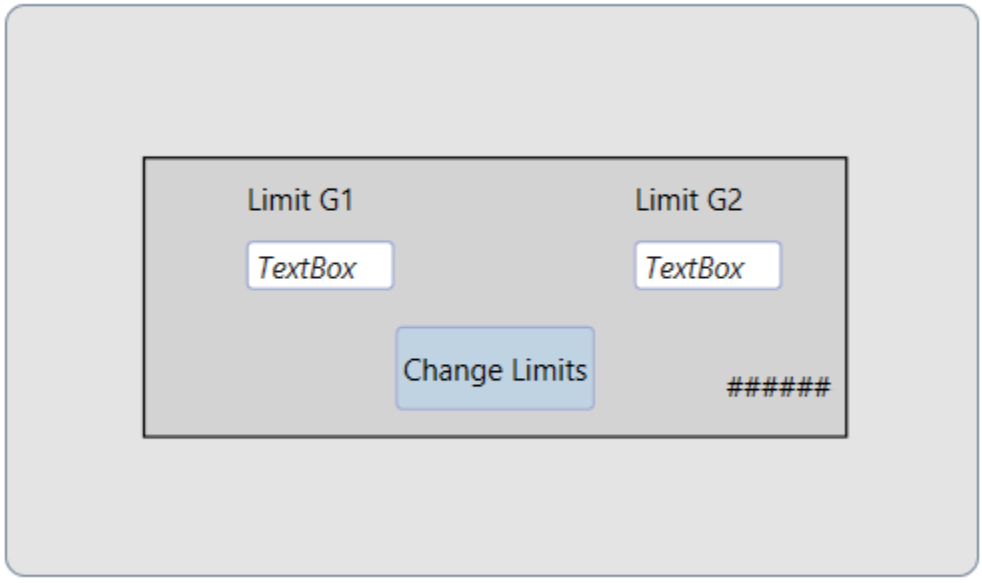
```
public String changeGroupsLimits()
{
	try
	{
		@Tag.Group1/Tag1.Hi.Limit = @Tag.LimitG1.ValueAsString;
		@Tag.Group1/Tag2.Hi.Limit = @Tag.LimitG1.ValueAsString;
		@Tag.Group1/Tag3.Hi.Limit = @Tag.LimitG1.ValueAsString;

		@Tag.Group2/Tag1.Hi.Limit = @Tag.LimitG2.ValueAsString;
		@Tag.Group2/Tag2.Hi.Limit = @Tag.LimitG2.ValueAsString;
		@Tag.Group2/Tag3.Hi.Limit = @Tag.LimitG2.ValueAsString;
		return "OK";
	}
	catch
	{
		return "ERROR";
	}

}
```

As explained before, "Hi" condition is a property of tag and the limit can be accessed by Tag.Group1_1.Hi.Limit, for example.

And finally, in Displays Draw > MainPage, we created controls to call the method "changeGroupsLimits". That way, when the button is pressed in runtime, the group limits will change.



In this section...
