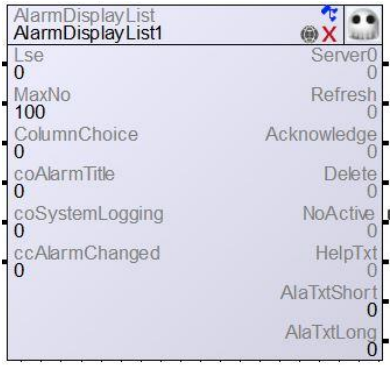
AlarmDisplayList



This class serves for building the current alarm list in a LASAL Screen project.

Additionally here the two alarm texts short and long for the alarm details are provided (also called alarm description).

Displaying the Alarm List in the Visualization

First a connection from the client Lse to a _Lse object has to be established, so that the class can access the kernel functions. To display the alarm list, the Server0 has to be placed in the LSE projects as NumEdit. In the LSE, the Virtual Objectname is the object name of the placed class (e.g. AlarmDisplayList1).

Additionally the client coSystemLogging connected to a SystemLogging object allows logging.

Defining the Columns

Here, the columns to be displayed in the visualization can be defined. The client ColumnChoice can be set as follows, where 1 = active:

- 2#1000 = alarm number
- 2#0100 = time received/cleared
- 2#0010 = cycle number ("Para1")
- 2#0001 = alarm text

Setting the Ringbuffer Size

With the client MaxNo the maximum number of entries in the ringbuffer can be defined. An entry is an alarm.

Interfaces

Clients

Lse	Object channel to the LSE object Data type: Object channel to the _Lse class
MaxNo	maximum number of entries in the ringbuffer Data type: UDINT
ColumnChoice	Bit pattern to enable the display of the display item: 2#1000 = alarm number 2#0100 = time received/cleared 2#0010 = cycle number (KaiAnd: real "Para1") 2#0001 = alarm text Data type: BDINT
coAlarmTitle	Object channel to AlarmTitleLineto influence the list header (optional) Data type: Object channel to the AlarmTitleLine class
coSystemLogging	Object channel to Logging function (optional) Data type: Object channel to the AlarmTitleLine class
ccAlarmChanged	Command channel to share alarm changes (ptional) Data type: DINT

Server

Server0

(Server0 inherited from _myIO)

Unit	-	Data type	DINT
Value Range	-	Write Protected	FALSE
Default value	-	Retentive	FALSE

Refresh

(Refresh inherited from _myIO)

Unit	-	Data type	DINT
Value Range	-	Write Protected	FALSE
Default value	-	Retentive	FALSE

Acknowledge

-1 = acknowledge all alarms
+x = acknowledge selected alarm

Unit	-	Data type	DINT
Value Range	-1 … x	Write Protected	FALSE
Default value	-	Retentive	FALSE

Delete

-1 = delete all alarms
+x = delete selected alarm

Unit	-	Data type	DINT
Value Range	-1 … x	Write Protected	FALSE
Default value	-	Retentive	FALSE

NoActive

Number of active alarms ... only after .Read()

Unit	-	Data type	UDINT
Value Range	-	Write Protected	TRUE
Default value	-	Retentive	FALSE

HelpTxt

Command “read alarm texts for alarm details” of the selected alarm

Unit	-	Data type	DINT
Value Range	-	Write Protected	FALSE
Default value	-	Retentive	FALSE

AlaTxtShort	Alarm text short after the read command

Unit	-	Data type	UDINT
Value Range	-	Write Protected	FALSE
Default value	-	Retentive	FALSE

AlaTxtLong	Alarm text long after the read command

Unit	-	Data type	UDINT
Value Range	-	Write Protected	FALSE
Default value	-	Retentive	FALSE

Globale Methoden

GetEvent	Is called each time, when the input is active and an event occurs: IN: **ped** pointer to _EDITOR information IN: **pe** pointer to _EVENT information OUT: **retcode** for the returnstatement there are 3 different possibilities _IDLE the system should go on like it is (system goes on) _IDIDIT i did it (system is ready) _IFAILED i tried to do my best, but i failed (system is ready)
IF_Start	method is called once right in front of drawing the object, for example open screen IN **pio** pointer to _IO information IN **firsttime** TRUE: when system wants to view drawing after open screen FALSE: when system wants to redraw background
IF_Run	method is called cyclic during object is on screen IN **pio** pointer to _IO information IN **input** TRUE it's an input FALSE it's an output
Line	method is called every time when redraw of a single line is necessary IN **ps** pointer to structure _SCROLL IN **pr** pointer to place where line should be drawn IN **line** number of line which should be drawn IN **state** line ia selected (TRUE) or passive (FALSE)

LineHeight	Outputs the double value of preselect are retcode IN **preselect** Preselection OUT **retcode** double value of preselect
SetAlarmInfo	Reads two alarm texts and writes them to the objects "AlarmTxtShort" and "AlarmTxtLong" / for the visualization) IN **psa** pointer to a SingleAlarm object