



# The AlarmTextAssigner Add-On Module for Saia PCD® Supervisor

<b>0</b>	<b>Contents</b>	<b>Page</b>
<b>0</b>	<b>Contents</b> .....	<b>1</b>
<b>1.</b>	<b>Document version</b> .....	<b>2</b>
<b>2.</b>	<b>Description</b> .....	<b>3</b>
<b>3.</b>	<b>Good to know</b> .....	<b>4</b>
<b>4.</b>	<b>Necessary conditions</b> .....	<b>4</b>
<b>5.</b>	<b>Preparations in PG5</b> .....	<b>5</b>
5.1.	FBoxes required.....	5
5.2.	DDC_SV_AlarmText.CSV.....	7
5.3.	Function of the AKS FBox.....	8
<b>6.</b>	<b>Preparations in the supervisor</b> .....	<b>12</b>
6.1.	Import of the relevant data points.....	12
6.2.	Settings in the AlarmTextAssigner tool.....	13
6.3.	Executing the tool.....	13
<b>7.</b>	<b>Connecting to a DDC Suite project</b> .....	<b>15</b>
<b>8.</b>	<b>Special features</b> .....	<b>16</b>

## 1. Document version

Date	Version	of	Description
2020-04-01	1.0	T. M.	First version
2020-04-27	1.1	T. M.	Revision of Section 2.5 and Chapter 6
2020-04-27	1.2	T. M.	<ul style="list-style-type: none"><li>• Minor corrections in Chap. 2</li><li>• Document layout</li></ul>
2020-04-28	1.3	T. M.	New Section 5.3 inserted. Corrections in Chap. 2
2020-05-03	1.4	T. M.	New Chapter 7 inserted.
2020-06-10	1.5	M. W.	Minor corrections
2020-06-23	1.6	M. H.	Takeover by SBC-TechDoc
2020-07-21	ENG01	M. H.	Translation into English

## 2. Description

This document describes the use of the AlarmTextAssigner tool. This small but efficient tool provides a simple way for not only new projects but also existing projects to integrate the alarm texts previously used in the Saia PG5 project (DDC\_Alarming.csv) into the Saia PCD Supervisor. The AlarmTextAssigner can be used with PG5 projects that were created with Advanced Alarming. The following FBoxes are used here:

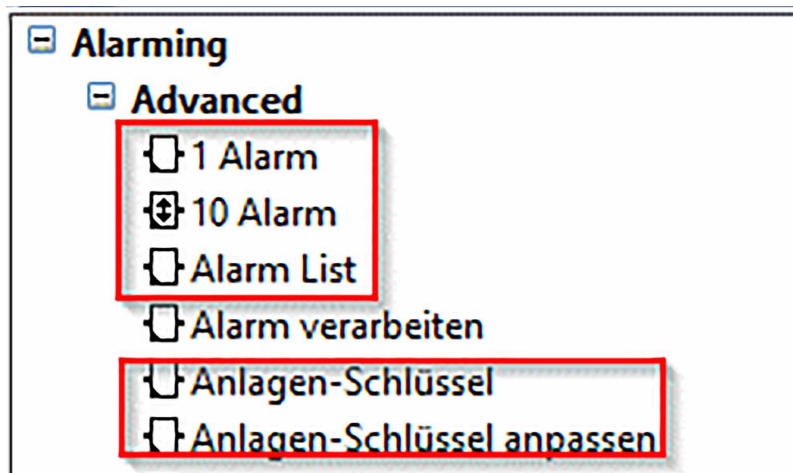


Figure 1: FBoxes from the Alarming library

After successful import and execution of the tool, an alarm extension with already filled alarm text is automatically attached to each configured alarm data point:

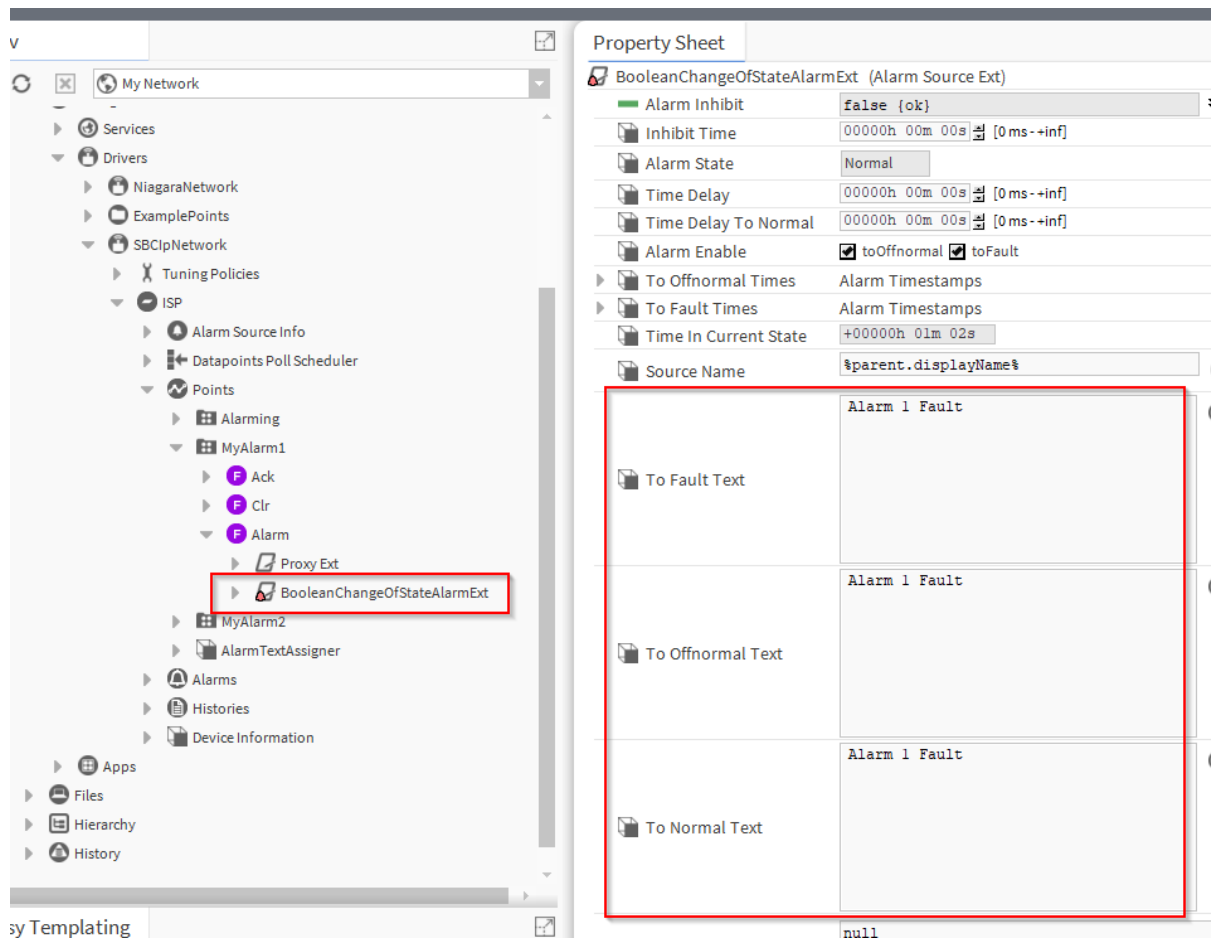


Figure 2: Example of an attached alarm extension

### **3. Good to know**

When using this tool, it is important to note that this type of alarm display only works in conjunction with MBE/GLT systems. The alarm interface of the PCD is not used here and therefore does not replace control of the alarms by the user/the PCD application.

### **4. Necessary conditions**

The module alarmTextAssigner-rt, version Honeywell 4.4.0.0.4, as well as the FBox library SWA\_Alarming version SP2.9.240 or better is required.

## 5. Preparations in PG5

### 5.1. FBoxes required

The FBoxes of Advanced Alarming can be used as required. An alarm list is therefore required, filled with alarms. This could for example look like the following:

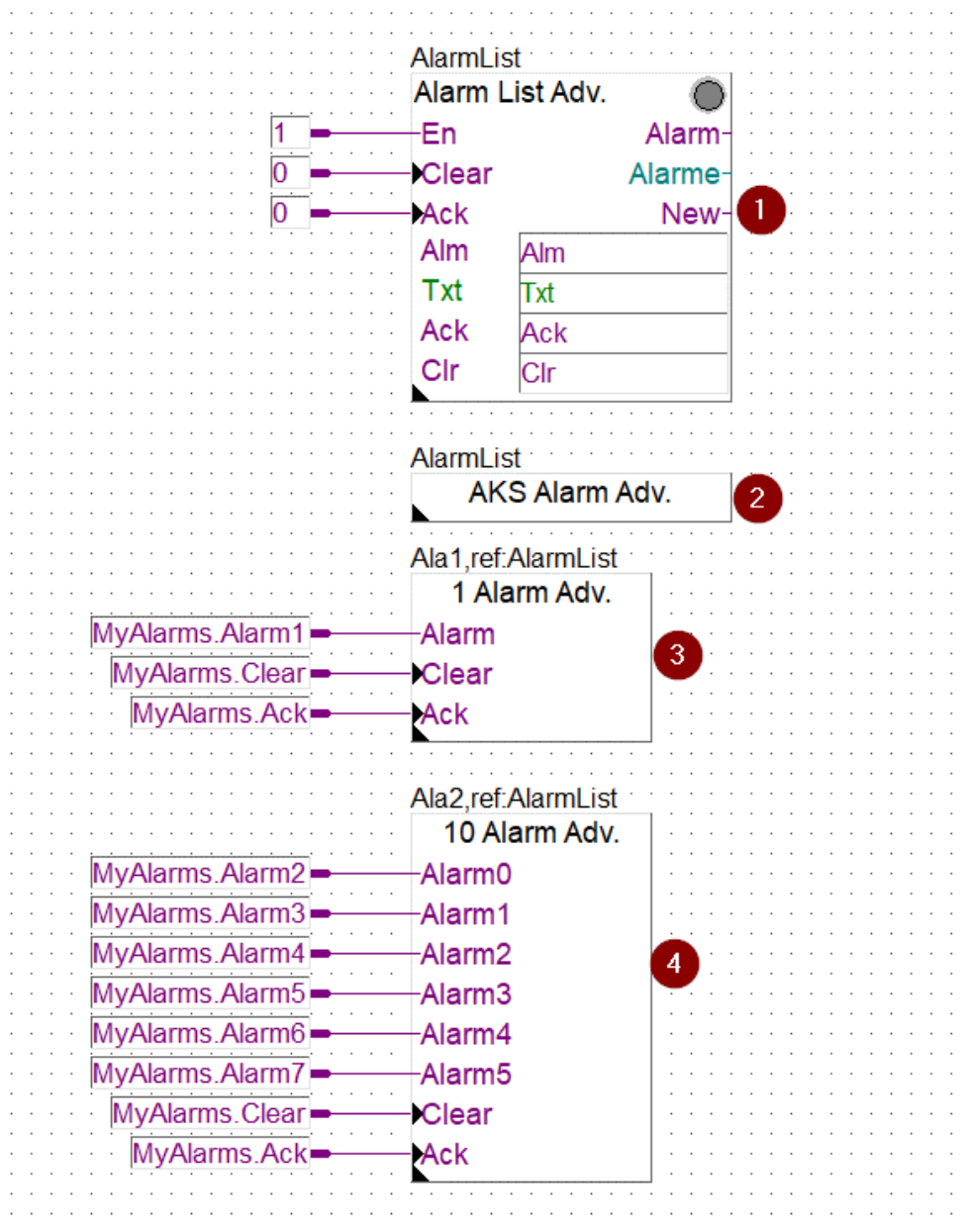



Figure 3: Example of an alarm program with Advanced Alarming

In the above example, the FBoxes from the Advanced (=Adv) family perform the following functions:

- Alarm List Adv. (1): FBox “Alarm List” is used to initialize the alarm list (AlarmList) and contains extensive additional functions.
- AKS Alarm Adv. (2): The FBox “Create system key” is used to generate a special system identification key (AKS)
- 1 Alarm Adv. (3): The FBox “1 Alarm” binds an alarm into the referenced alarm list (AlarmList)
- 10 Alarm Adv. (4): The FBox “10 Alarm” binds up to 10 alarms into the referenced alarm list (AlarmList)

	<b>Important:</b>	<ul style="list-style-type: none"> <li>• The input connectors must have symbols assigned.</li> <li>• No negator may be set on the symbol or on the FBox input. If a negation should be carried out, the negator must be placed elsewhere (for example where the symbol is passed from another FBox).</li> </ul>
---	-------------------	---

If there are free alarms in the project anyway, they can be implemented as follows:

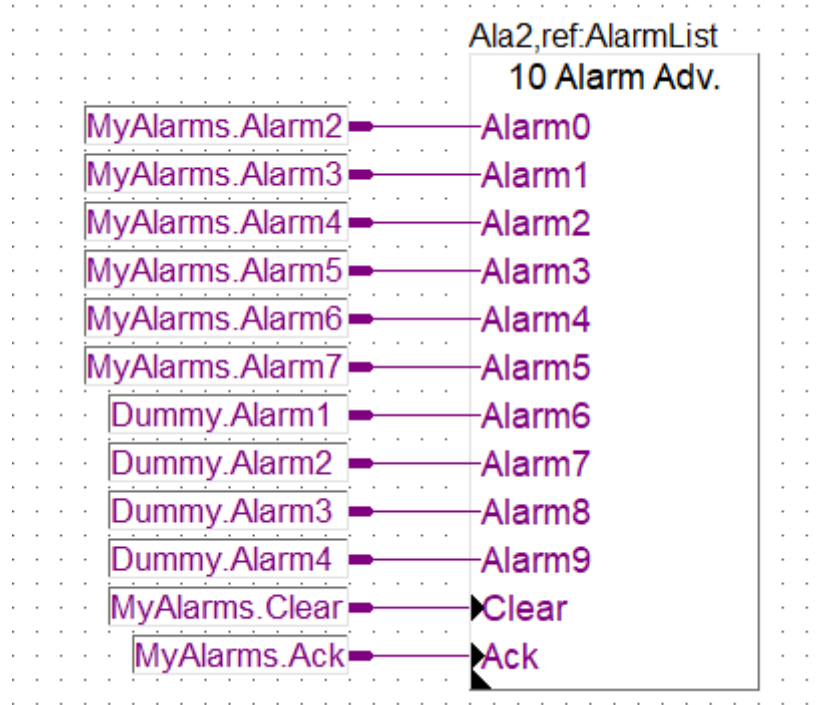


Figure 4: Example of the handling of reserve alarms

Alternatively, the FBox can also be pulled upwards only as far as needed.

## 5.2. DDC\_SV\_AlarmText.CSV

With the FBox library SWA\_Alarming, version SP2.9.240 or better, a file “DDC\_SV\_AlarmText.CSV” is automatically created under the PG5 project in the corresponding device folder. In the example in Figure 3, this file looks like the following:

MyAlarms.Alarm1	Alarm 1 Fault	Alarm 1 Fault	Alarm 1 Fault	AlarmClass='Prio_1'
MyAlarms.Alarm2	Sys.1-Alarm x	Sys.1-Alarm x	Sys.1-Alarm x	AlarmClass='Prio_1'
MyAlarms.Alarm3	Sys.1-Alarm x+1	Sys.1-Alarm x+1	Sys.1-Alarm x+1	AlarmClass='Prio_1'
MyAlarms.Alarm4	Sys.1-Alarm x+2	Sys.1-Alarm x+2	Sys.1-Alarm x+2	AlarmClass='Prio_1'
MyAlarms.Alarm5	Sys.1-Alarm x+3	Sys.1-Alarm x+3	Sys.1-Alarm x+3	AlarmClass='Prio_1'
MyAlarms.Alarm6	Sys.1-Alarm x+4	Sys.1-Alarm x+4	Sys.1-Alarm x+4	AlarmClass='Prio_1'
MyAlarms.Alarm7	Sys.1-Alarm x+5	Sys.1-Alarm x+5	Sys.1-Alarm x+5	AlarmClass='Prio_1'

Figure 5: Example of the CSV file generated

Only Alarm1 receives a different text from its FBox:

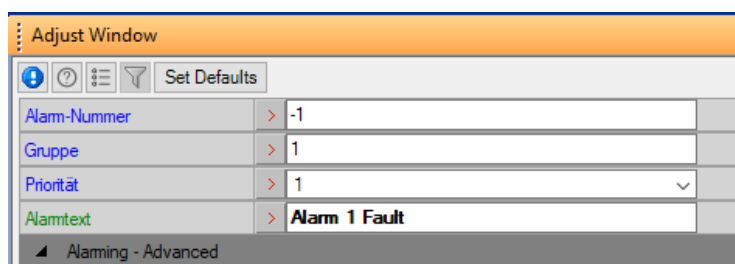


Figure 6: 1 Alarm FBox settings

Alarms 2 through 7 are given the default values:

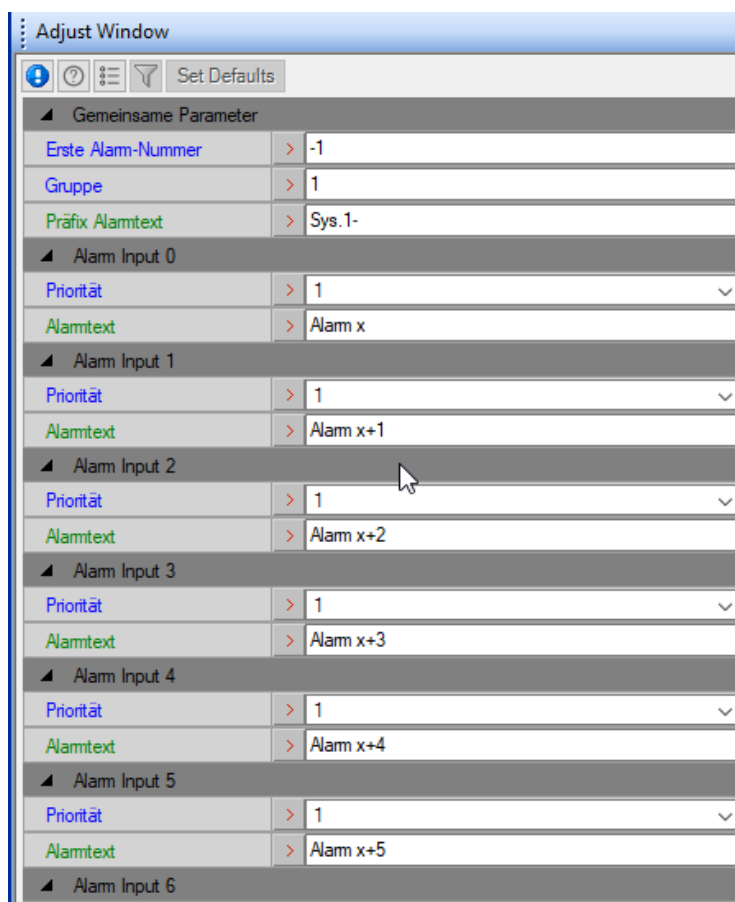


Figure 7: 10 Alarm FBox settings

The alarm texts can be freely selected, as can the AKS identifiers.

### 5.3. Function of the AKS FBox

In the example shown above, the FBox handles the assignment of priorities to the alarm classes of the supervisor. If an AKS will also be used, this can be implemented as follows:

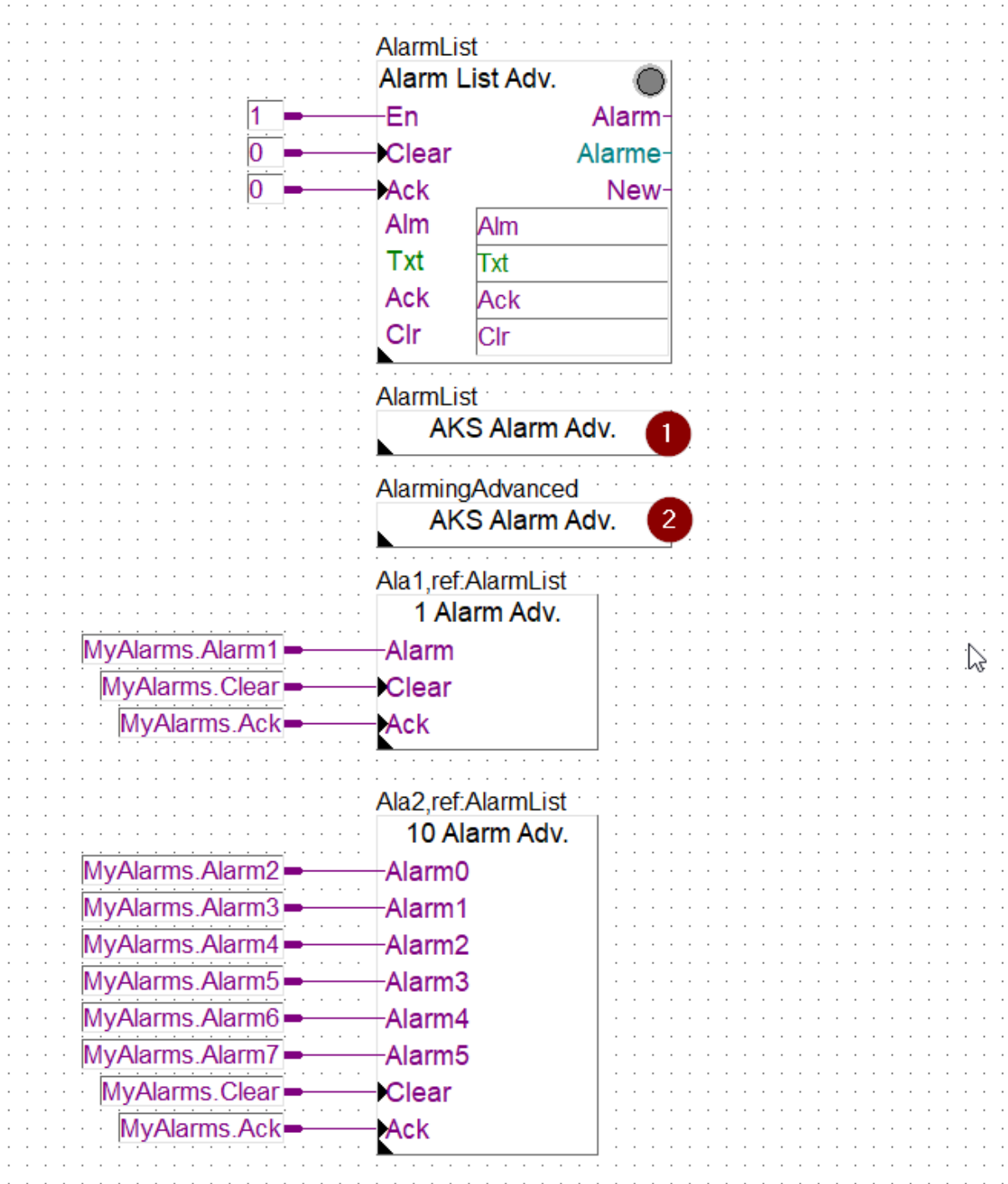


Figure 8: Example of working with an AKS



The two AKS FBoxes (1 and 2) handle different functions here. The first FBox defines the supervisor alarm classes:

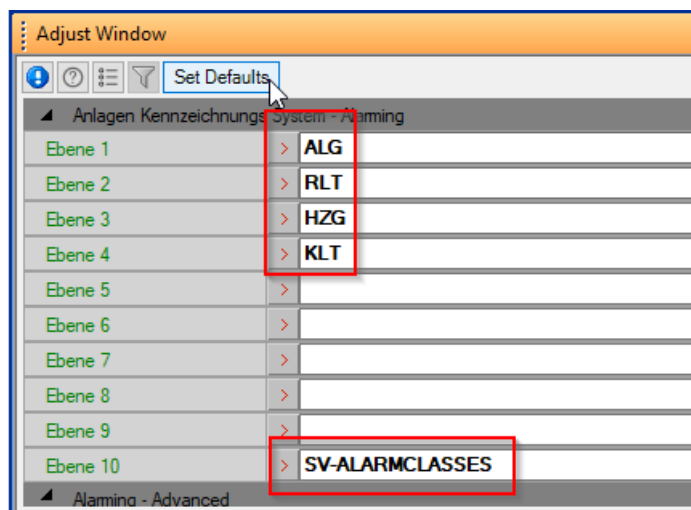


Figure 9: AKS FBox with assignment of the alarm classes

Column 5 in DDC\_SV\_AlarmText.CSV automatically receives the extension AlarmClass'Prio\_1' from the associated priority. If the AlarmClass is named differently, the AKS FBox must be adapted and the extension "SV-ALARMCLASSES" added to level 10 (Figure 9). The entry SV-ALARMCLASSES refers to up to 7 priorities (4 were defined in the figure above). This entry tells the FBox that this is an extension to the supervisor alarm. After assigning the alarms to a priority, after recompiling the DDC\_SV\_AlarmText.CSV is adapted to:

	A	B	C	D	E
1	MyAlarms.Alarm1	Alarm 1 Fault	Alarm 1 Fault	Alarm 1 Fault	AlarmClass="ALG"
2	MyAlarms.Alarm2	Sys.1-Alarm x	Sys.1-Alarm x	Sys.1-Alarm x	AlarmClass="ALG"
3	MyAlarms.Alarm3	Sys.1-Alarm x+1	Sys.1-Alarm x+1	Sys.1-Alarm x+1	AlarmClass="ALG"
4	MyAlarms.Alarm4	Sys.1-Alarm x+2	Sys.1-Alarm x+2	Sys.1-Alarm x+2	AlarmClass="RLT"
5	MyAlarms.Alarm5	Sys.1-Alarm x+3	Sys.1-Alarm x+3	Sys.1-Alarm x+3	AlarmClass="RLT"
6	MyAlarms.Alarm6	Sys.1-Alarm x+4	Sys.1-Alarm x+4	Sys.1-Alarm x+4	AlarmClass="HZG"
7	MyAlarms.Alarm7	Sys.1-Alarm x+5	Sys.1-Alarm x+5	Sys.1-Alarm x+5	AlarmClass="HZG"

Figure 10: DDC\_SV\_AlarmText.CSV with alarm classes

The second FBox in Figure 8 is then used to generate the AKS:

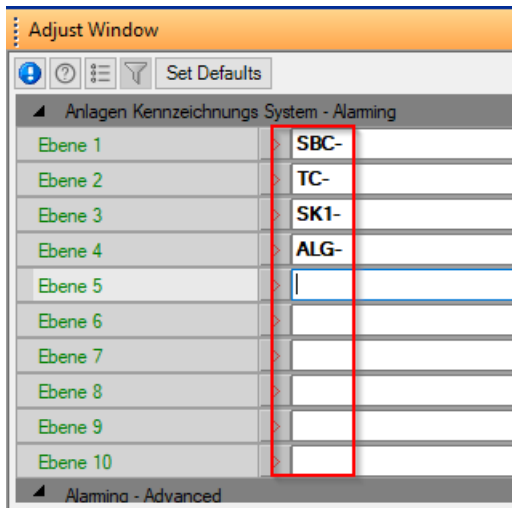


Figure 11: AKS FBox with AKS assignment

After compiling again, the AKS is added to the alarms and the DDC\_SV\_AlarmText.CSV is updated as follows:

	A	B	C	D	E
1	MyAlarms.Alarm1	SBC-TC-SK1-ALG-Alarm 1 Fault	SBC-TC-SK1-ALG-Alarm 1 Fault	SBC-TC-SK1-ALG-Alarm 1 Fault	AlarmClass='ALG'
2	MyAlarms.Alarm2	SBC-TC-SK1-ALG-Sys.1-Alarm x	SBC-TC-SK1-ALG-Sys.1-Alarm x	SBC-TC-SK1-ALG-Sys.1-Alarm x	AlarmClass='ALG'
3	MyAlarms.Alarm3	SBC-TC-SK1-ALG-Sys.1-Alarm x+1	SBC-TC-SK1-ALG-Sys.1-Alarm x+1	SBC-TC-SK1-ALG-Sys.1-Alarm x+1	AlarmClass='ALG'
4	MyAlarms.Alarm4	SBC-TC-SK1-ALG-Sys.1-Alarm x+2	SBC-TC-SK1-ALG-Sys.1-Alarm x+2	SBC-TC-SK1-ALG-Sys.1-Alarm x+2	AlarmClass='RLT'
5	MyAlarms.Alarm5	SBC-TC-SK1-ALG-Sys.1-Alarm x+3	SBC-TC-SK1-ALG-Sys.1-Alarm x+3	SBC-TC-SK1-ALG-Sys.1-Alarm x+3	AlarmClass='RLT'
6	MyAlarms.Alarm6	SBC-TC-SK1-ALG-Sys.1-Alarm x+4	SBC-TC-SK1-ALG-Sys.1-Alarm x+4	SBC-TC-SK1-ALG-Sys.1-Alarm x+4	AlarmClass='HZG'
7	MyAlarms.Alarm7	SBC-TC-SK1-ALG-Sys.1-Alarm x+5	SBC-TC-SK1-ALG-Sys.1-Alarm x+5	SBC-TC-SK1-ALG-Sys.1-Alarm x+5	AlarmClass='HZG'

Figure 12: DDC\_SV\_AlarmText.CSV with AKS and alarm classes

If the AKS is to be adapted in parts of the Fupla program, the AKS Alarm Adv FBoxes can be used (Figure 13):

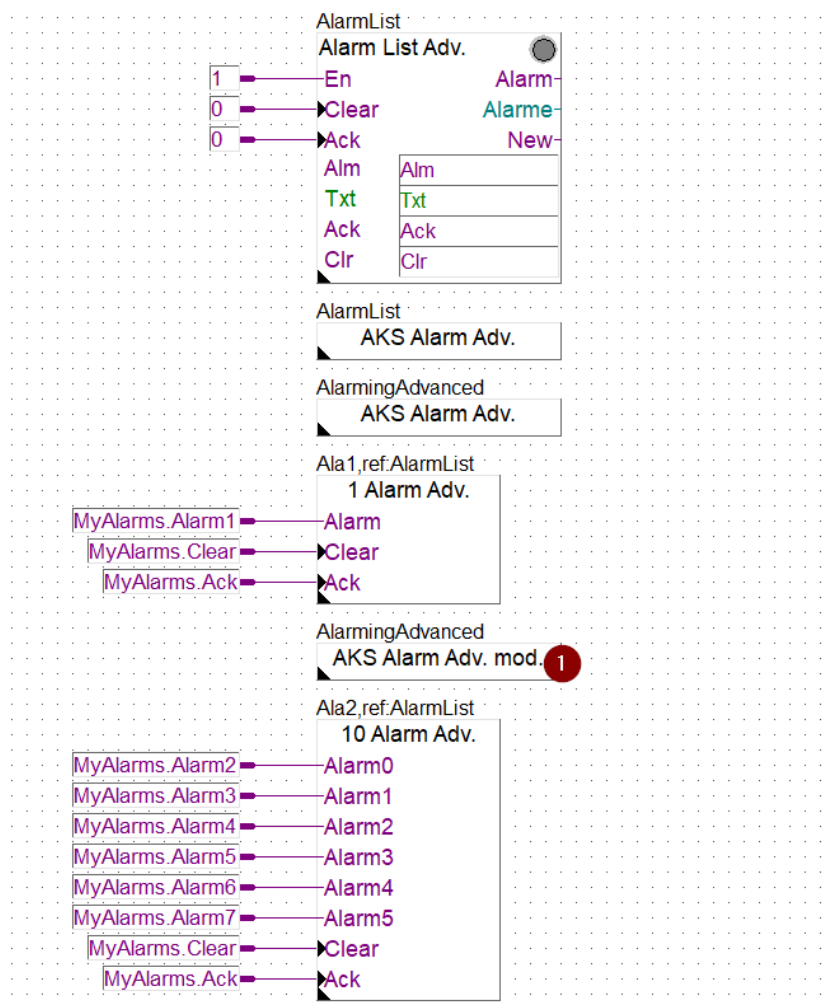


Figure 13: Adaptation of the AKS in the Fupla program

These FBoxes adjust the AKS before each Alarm FBox set. In the following example (Figure 14), level 4 has been adjusted:

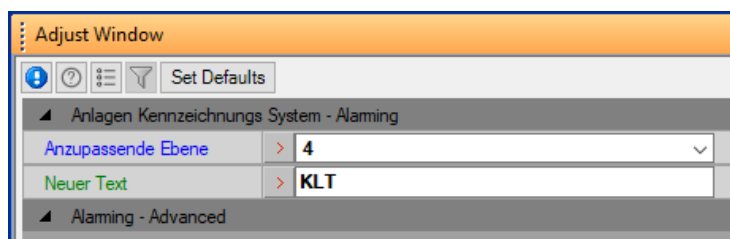


Figure 14: Modifying the AKS

After the compilation process, DDC\_SV\_AlarmText.CSV changes to:

	A	B	C	D	E
1	MyAlarms.Alarm1	SBC-TC-SK1-ALG Alarm 1 Fault	SBC-TC-SK1-ALG Alarm 1 Fault	SBC-TC-SK1-ALG Alarm 1 Fault	AlarmClass="ALG"
2	MyAlarms.Alarm2	SBC-TC-SK1-KLT Sys.1-Alarm x	SBC-TC-SK1-KLT Sys.1-Alarm x	SBC-TC-SK1-KLT Sys.1-Alarm x	AlarmClass="ALG"
3	MyAlarms.Alarm3	SBC-TC-SK1-KLT Sys.1-Alarm x+1	SBC-TC-SK1-KLT Sys.1-Alarm x+1	SBC-TC-SK1-KLT Sys.1-Alarm x+1	AlarmClass="ALG"
4	MyAlarms.Alarm4	SBC-TC-SK1-KLT Sys.1-Alarm x+2	SBC-TC-SK1-KLT Sys.1-Alarm x+2	SBC-TC-SK1-KLT Sys.1-Alarm x+2	AlarmClass="RLT"
5	MyAlarms.Alarm5	SBC-TC-SK1-KLT Sys.1-Alarm x+3	SBC-TC-SK1-KLT Sys.1-Alarm x+3	SBC-TC-SK1-KLT Sys.1-Alarm x+3	AlarmClass="RLT"
6	MyAlarms.Alarm6	SBC-TC-SK1-KLT Sys.1-Alarm x+4	SBC-TC-SK1-KLT Sys.1-Alarm x+4	SBC-TC-SK1-KLT Sys.1-Alarm x+4	AlarmClass="HZG"
7	MyAlarms.Alarm7	SBC-TC-SK1-KLT Sys.1-Alarm x+5	SBC-TC-SK1-KLT Sys.1-Alarm x+5	SBC-TC-SK1-KLT Sys.1-Alarm x+5	AlarmClass="HZG"
8					

Figure 15: Adapted AKS in DDC\_SV\_AlarmText.CSV

## 6. Preparations in the supervisor

### 6.1. Import of the relevant data points

First all the relevant data points are imported with the Saia PG5 Import Wizard. The data points initially have no alarm extension:

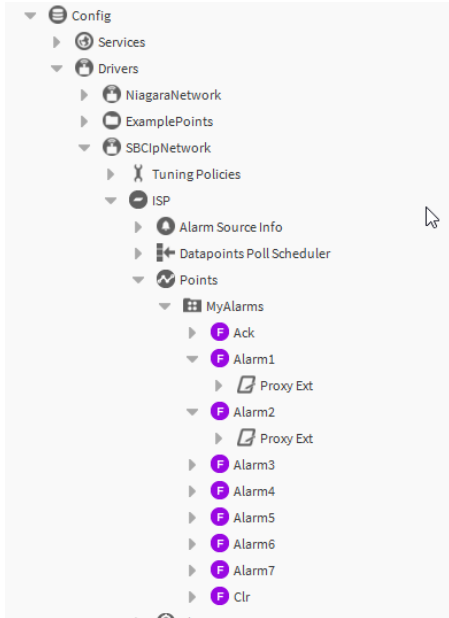


Figure 16: Alarms directly after import

To connect the Alarm Extension, drag the AlarmTextAssigner tool from the palette of the same name to the Points folder:

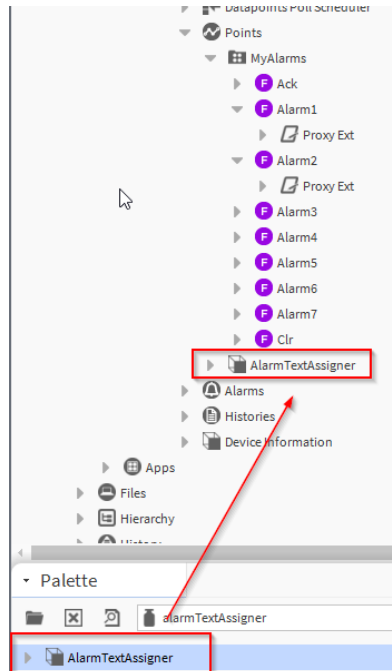


Figure 17: Adding the AlarmTextAssigner tool from the palette

The Points folder is an example here. For example, it can also be stored directly under the driver. However, it is recommended that you store it in the corresponding Device, otherwise the settings (see Section 6.2) would always have to be adjusted.

	<b>Important:</b>	It is no longer necessary to import the alarm list during the import process in the Saia PG5 Import Wizard!
--	-------------------	---

## 6.2. Settings in the AlarmTextAssigner tool

In the AlarmTextAssigner tool, a few settings must be configured before use:

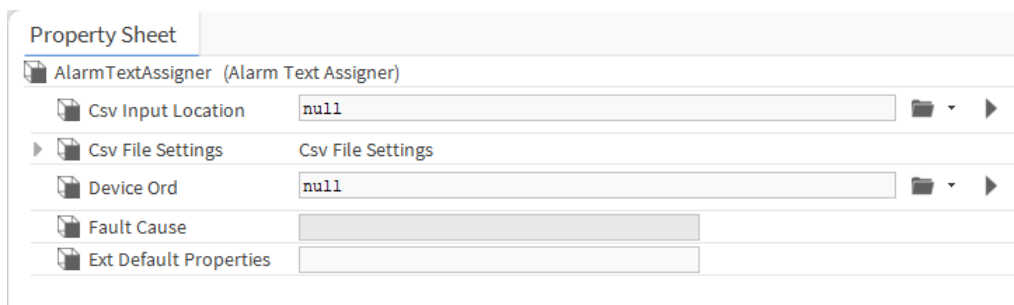


Figure 18: Default view

- Csv Input Location: Specifies the path to the file DDC\_SV\_AlarmText.CSV
- Csv File Settings: Provides additional optional settings (Delimiter, Default Header, Code Page)
- Device Ord: Specifies the path to the device in the driver
- Fault Cause: Specifies any errors
- Ext Default Properties: Additional parameters can be passed here in the same way as in PointDefinition.csv (separate entries with “;”). This could for example be:
  - SourceName='@Path@ test'
  - AlarmClass='AlarmClass\_Prio1'

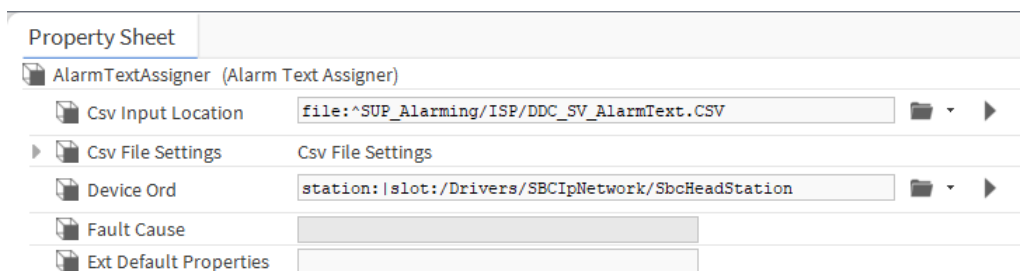


Figure 19: Completely configured tool

## 6.3. Executing the tool

The tool can be run directly from the property sheet

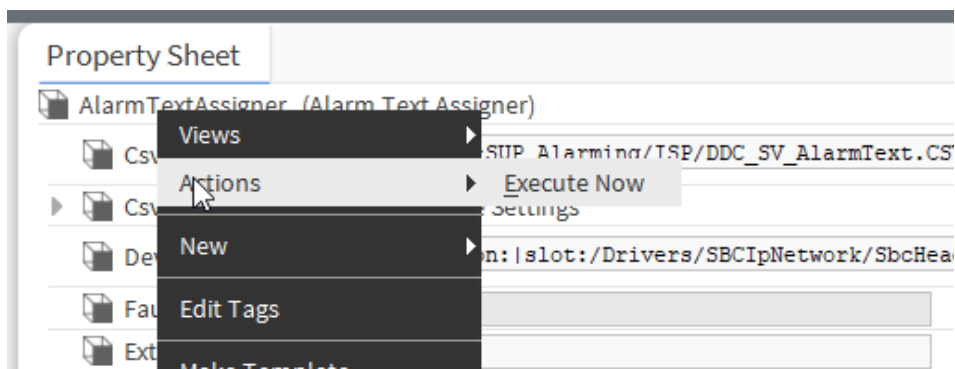


Figure 20: Execution directly from the property sheet

or from the NavTree:

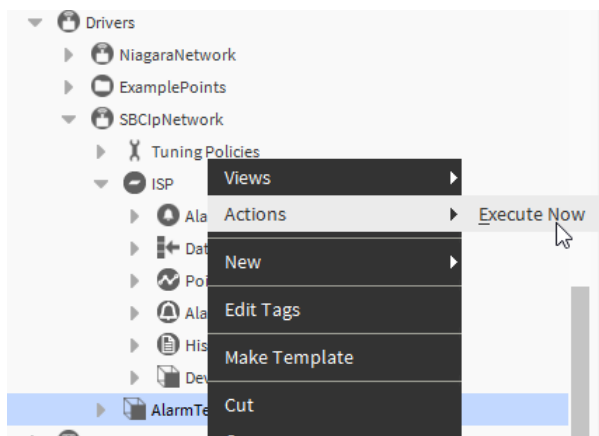


Figure 21: Execution from the NavTree

After the tool has been executed, a confirmation is displayed:

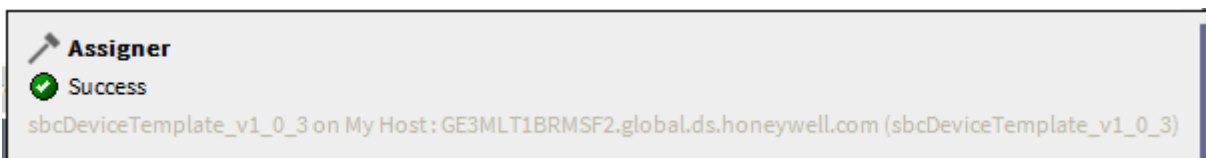


Figure 22: Confirmation of execution

The data points now have an alarm extension with the text configured in PG5:

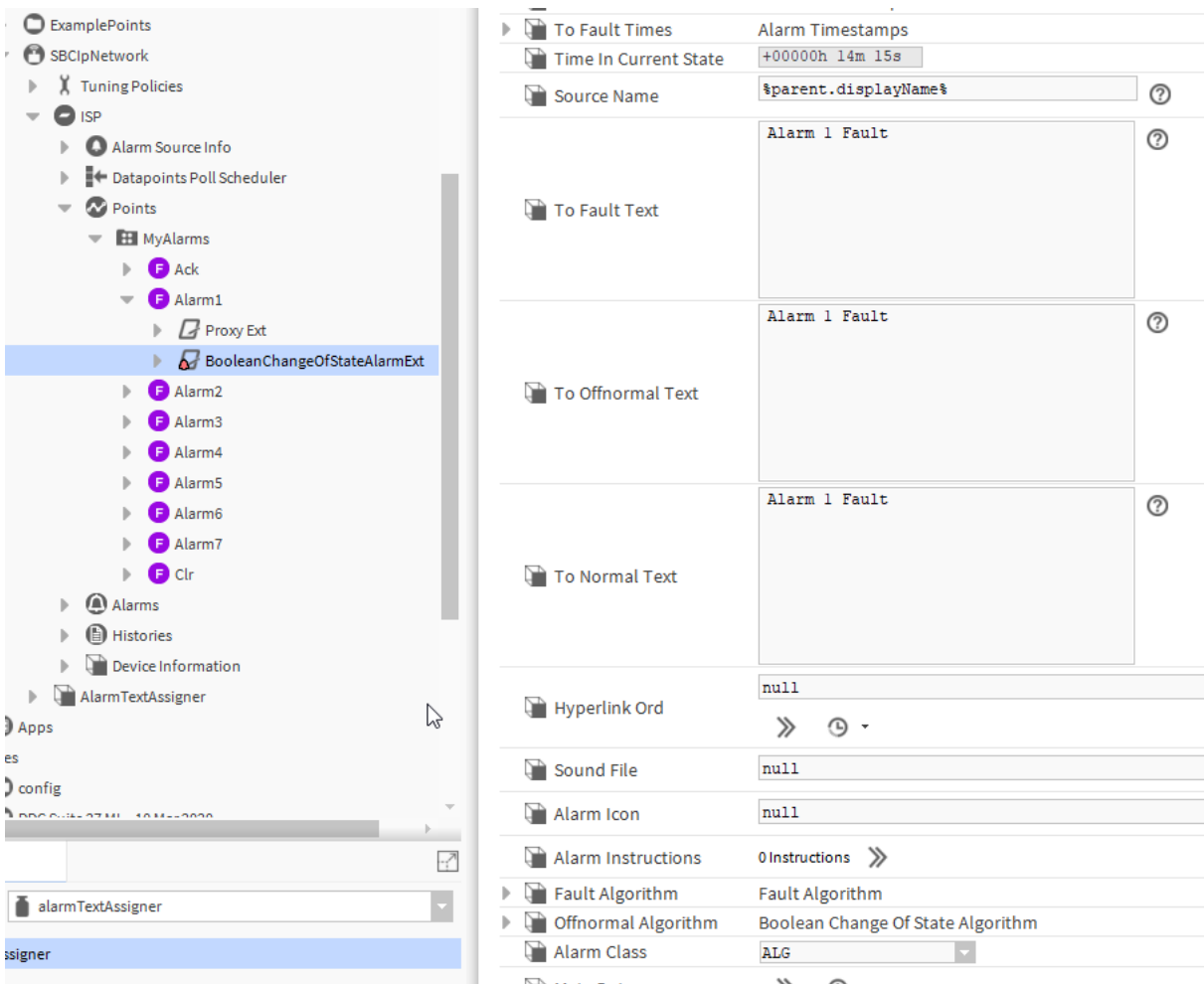


Figure 23: Alarm texts imported as Alarm Extension

## 7. Connecting to a DDC Suite project

If a DDC Suite project (with PointDefinition.csv included) has already been imported into the supervisor, by default the alarm extensions are created with a source text and an alarm call assignment.

That is, without alarm texts in the areas To Fault Text, To Offnormal Text and To Normal Text:

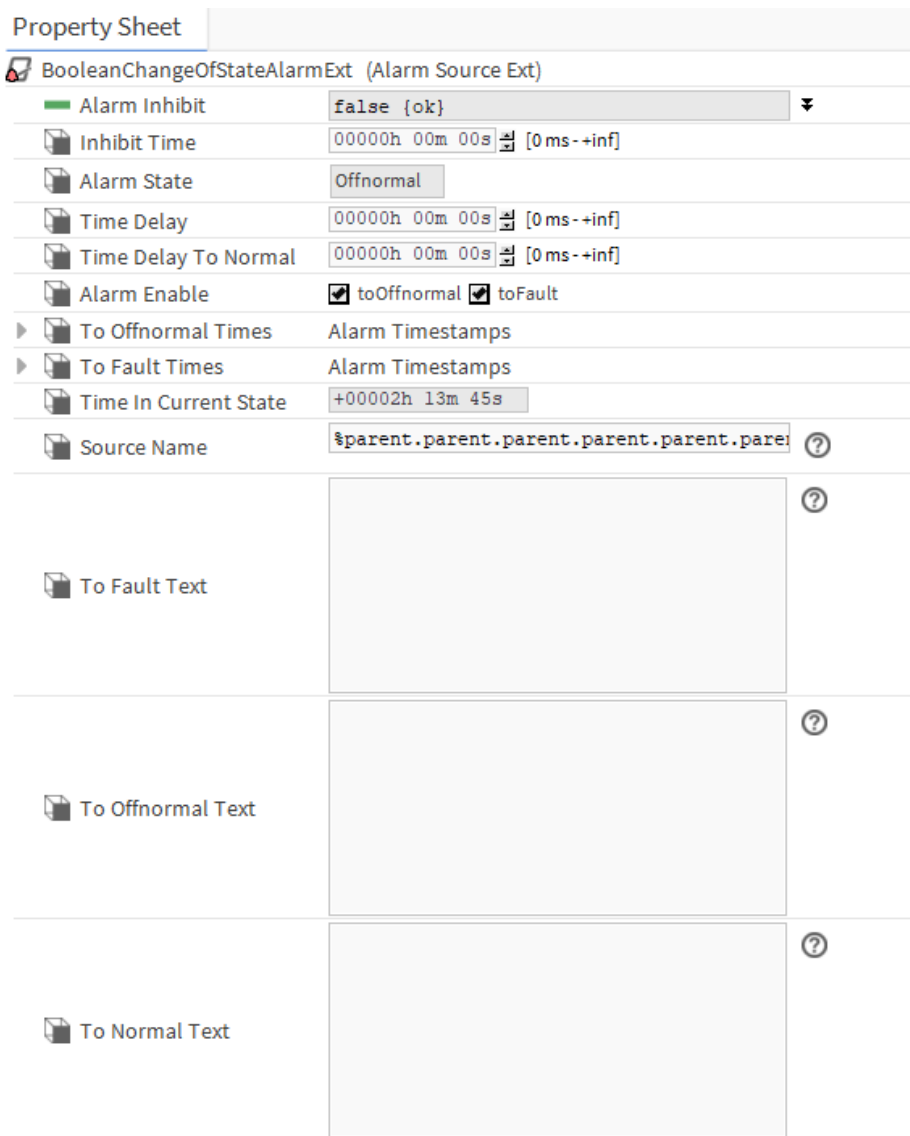


Figure 24: Property Sheet

During the build of the DDC Suite project, a “DDC\_SV\_AlarmText.CSV” file as described in Section 5.2 was already created. In connection with the DDC Suite Alarming, however, the content is a bit less, because the alarm texts and parts of the AKS are created directly in the DDC Suite FBoxes. In addition, the alarm class and source name are already created by the “PointDefinition.csv” during the process of importing into the supervisor.

Without manual adaptation, the file initially looks like this:

	A	B	C	D	E
1	Init.Library.FBox.PcdError	SBC TC PCD Internal Error	SBC TC PCD Internal Error	SBC TC PCD Internal Error	
2	Init.Library.FBox.SmBatterie	SBC TC PCD Battery Weak	SBC TC PCD Battery Weak	SBC TC PCD Battery Weak	
3	Alarming.Voltages.FBox.AlaAC230	SBC TC PCD 230VAC	SBC TC PCD 230VAC	SBC TC PCD 230VAC	
4	Alarming.Voltages.FBox.AlaAC24	SBC TC PCD 24VAC	SBC TC PCD 24VAC	SBC TC PCD 24VAC	
5	Alarming.Voltages.FBox.AlaDC24	SBC TC PCD 24VDC	SBC TC PCD 24VDC	SBC TC PCD 24VDC	
6	Alarming.Voltages.FBox.AlaPhase	SBC TC PCD Phase Monitoring	SBC TC PCD Phase Monitoring	SBC TC PCD Phase Monitoring	
7	Alarming.Voltages.FBox.AlaVoltage	SBC TC PCD Main Voltage	SBC TC PCD Main Voltage	SBC TC PCD Main Voltage	
8	Alarming.Alarm1.FBox.Ala	SBC TC PCD AlarmAlarm	SBC TC PCD AlarmAlarm	SBC TC PCD AlarmAlarm	
9	Alarming.AlarmMessage.FBox.Ala	SBC TC PCD AlarmAlarm	SBC TC PCD AlarmAlarm	SBC TC PCD AlarmAlarm	

Figure 25: PointDefinitions

The Alarm Text Assigner is now added, configured and executed as described in Sections 6.2 and 6.3. After that, the texts are added in the areas “To Fault Text”, “To Offnormal Text” and “To Normal Text” in the appropriate extensions (the texts can be customized!):



Figure 26: Extension

## 8. Special features

- The document was produced for version Honeywell 4.4.0.0.4 of the tool
- Windows 10 Enterprise, 64 Bit, Version 1803

### Saia-Burgess Controls AG

Bahnhofstrass 18 | 3280 Murten, Switzerland  
T +41 26 580 30 00 | F +41 26 580 34 99  
www.saia-pcd.com

support@saia-pcd.com | www.sbc-support.com

2020-07-21 | Technical data and specifications subject to change.