

# **Configuration of Alarm System**

## **Process Module**

config\_alm\_process.dll

Written by: Charlotte Bell, Richard Kerry

# Contents

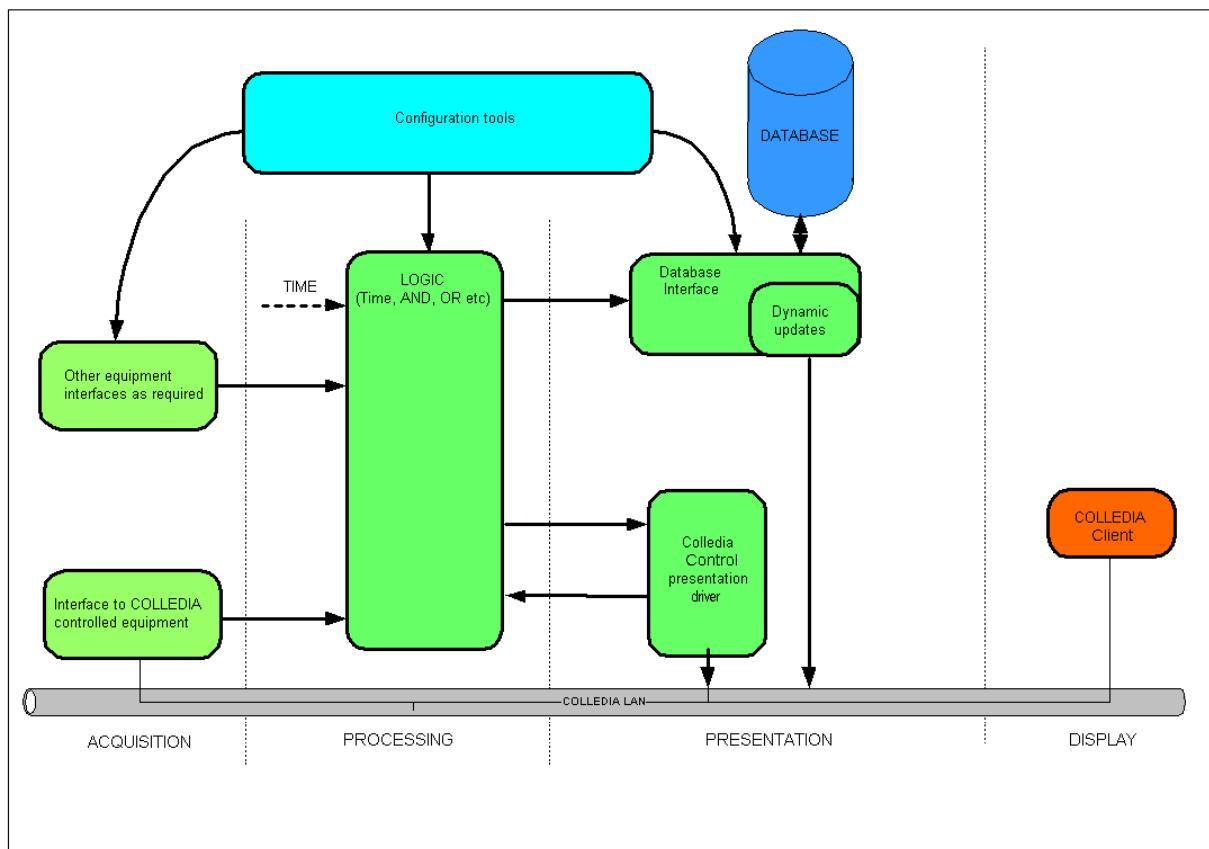
<b>Contents .....</b>	<b>2</b>
<b>1 Alarm System – Process Configuration .....</b>	<b>3</b>
1.1 Overview .....	3
1.2 Description .....	3
<b>2 Configuration Procedure .....</b>	<b>4</b>
2.1 Configuring the Processes .....	4
2.2 General Notes.....	5
2.3 Logic Functions .....	6
2.4 Properties Available for all Processes .....	10
2.5 Duplicating Processes.....	12
<b>3 Naming Convention.....</b>	<b>15</b>
<b>4 Documents Referenced.....</b>	<b>15</b>
<b>5 Version history.....</b>	<b>15</b>
5.1 Software Version.....	15
5.2 Document Version .....	16

# 1 Alarm System – Process Configuration

## 1.1 Overview

The logic module takes the input processes, performs the operations defined in the processing configuration and passes the results to the output modules. This document describes how to configure the processes in a typical system – as seen in the logic block in the diagram below.

The following is a block diagram of the Alarm System - the logic block is built in to the main application but needs to be configured in order to do anything useful.



## 1.2 Description

The logic module takes the input processes, performs the operations defined in the processing configuration and passes the results to the output modules. These operations can be logical operations

(OR, AND, NOT or TRUTH). In addition there are time of day functions and holdoffs.

Any process may be masked by another.

## 2 Configuration Procedure

To configure the processes the BNCS v4.5 configuration tool should be used. The following explanation assumes that you are running the configuration editor and that this is connected to the relevant configuration server. It also assumes that you have already configured some inputs in the Alarm System.

The Current System will show “select a system” if you have not yet selected one, in which case then select the Change System button, then select an alarm system from the drop down list. This step is needed because there may be several alarm systems within one BNCS installed system.

There is only one process configuration file (proc\_def.cfg) per Alarm System.

### 2.1 Configuring the Processes

#### 2.1.1 Input Processes

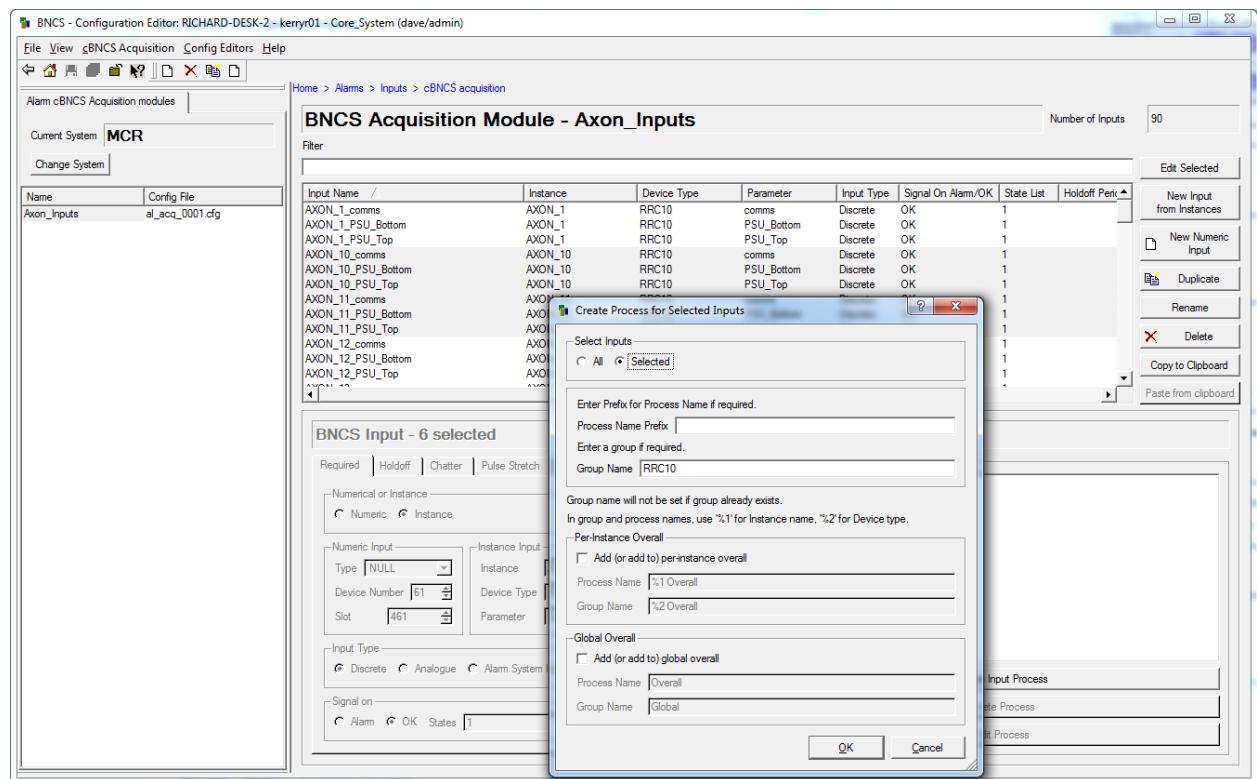
The first step is to define some input processes. For most systems these will correspond to all of the inputs created using the acquisition module configuration tools. The example below shows how to create input processes from the BNCS inputs already created. Using the BNCS Acquisition tool it is possible to edit the name, add a prefix to the name and associate different processes with different group names. It is advisable to group the processes according to their use as the processes can be sorted by this group.

Using the BNCS Acquisition Module Configuration tool select all the processes that are to be in one group, as shown. Press the “Create Input Process” button. A popup dialog appears.

All inputs may have processes created, or just those selected.

There are options to add a prefix to the input name and to associate these processes with a group name. We will not add any prefix, because the input names are already sensible. We will associate the group “SIM10” with our inputs. Press OK to create the processes.

It is also possible to automatically add the new processes to “overall” OR processes, based on each process’s instance name or device type. If these options are selected the new processes will be created if not already present.

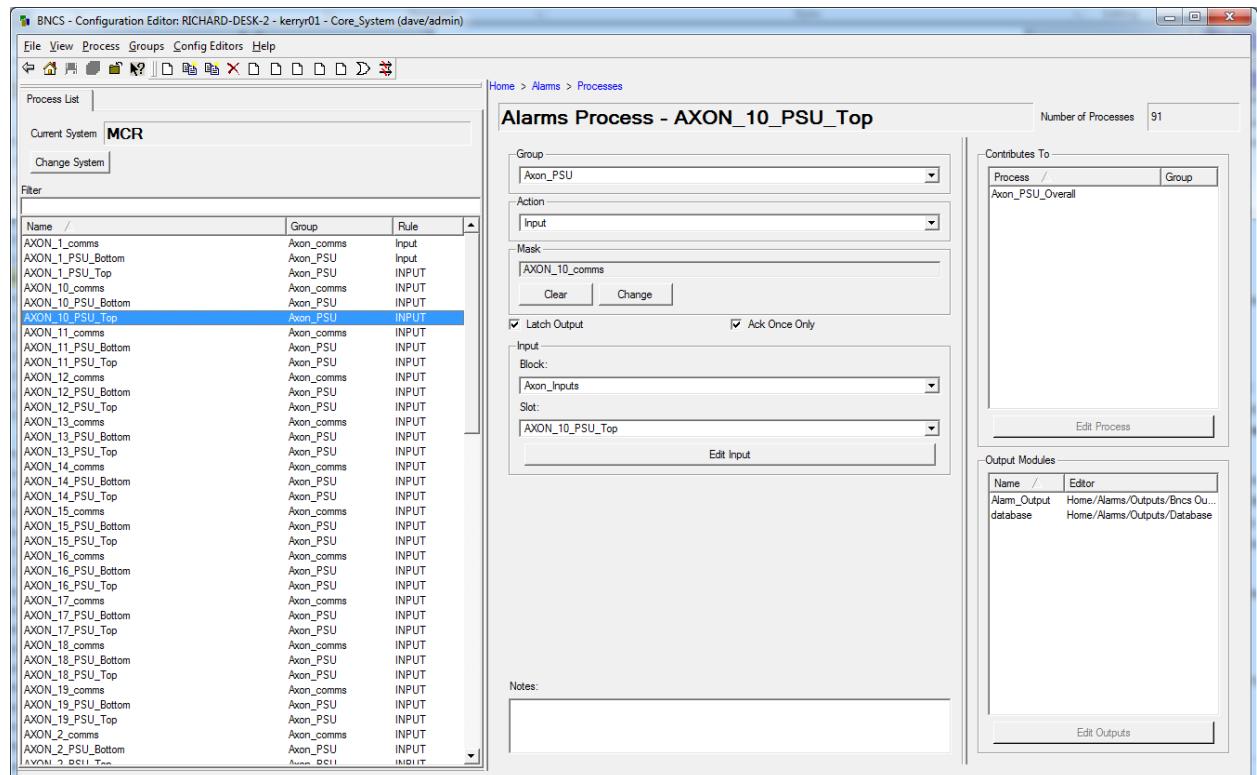


We now have some input processes. From the BNCS Acquisition configuration tool it is also possible to delete processes and also to directly navigate to the Alarms Process Configuration editor. For the rest of the configuration we will be using the Alarms Processes Configuration editor.

Select Alarms, then Processes.

## 2.2 General Notes

The screenshot below shows a configuration with all the processes listed in the bottom left hand pane. Highlighting any one process will show its details in the right hand pane. For example, the process highlighted is "AXON\_10\_PSU\_Top", it is in group "Axon\_PSU" and it is an "INPUT" process. The right hand pane also tells us which input module it comes from and which input it is. On the right hand side we can also see that this process contributes to another process called "Axon\_PSU\_Overall". From this screen it is also possible to navigate to the input configuration screen to look at the input and to navigate to the output configuration screens.



## 2.3 Logic Functions

### 2.3.1 OR Process - Grouping alarms

Simple aggregates of a group of alarms, such as all those from one device, may be grouped using an OR process. The output will be the same as the highest of the following levels of the inputs: -

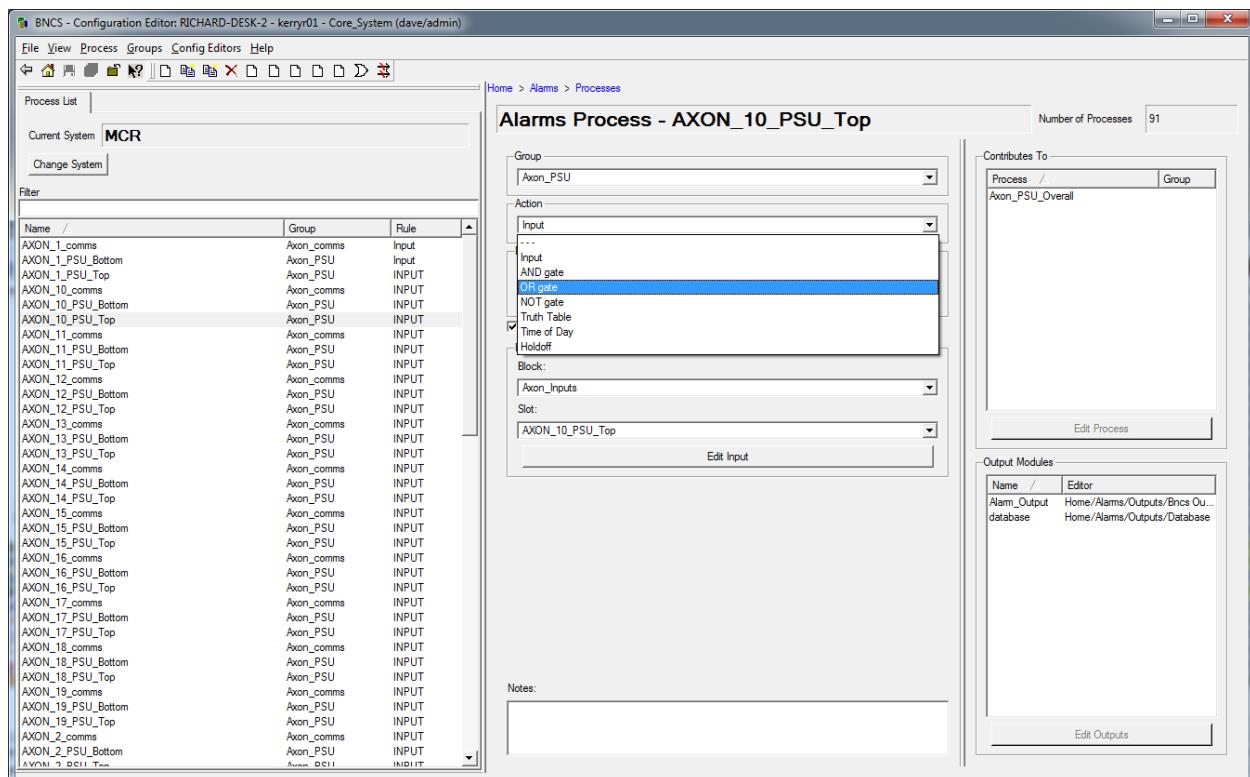
On/Force On (appears as Alarm)

Latched

Acknowledged

Off/Ignored/ Force off (appears as OK)

If you have some alarms that all need to contribute to the same OR process you will need to create a new process. You may either create a new process or duplicate an existing one. Either way, rename it and set its rule to "OR". The drop down list under the action label lists all of the types of process available.



In order to add the inputs to the process (or change them) press the “Change Inputs” button. Select the inputs required into or out of the right hand pane of the popup using the arrows. Select “OK” to assert the new settings.

An OR process has an option to re-trigger after it has been acknowledged if one of its inputs subsequently goes into the Alarm state. To configure this check the “Re-trigger on new error” box.

An additional option is for an OR process to treat latched or acknowledged inputs as if they are off. This can be useful when you need a sounder to latch and the inputs to latch, acknowledging the sounder will then go to OK rather than stay at latched because an input is latched. This setting is sometimes referred to as “ignore intermediate states”.

Note: it is important to bear in mind which processes will be presented to the panels and if you intend them to be overridden (i.e. acknowledged, forced off etc.). This is because an override affects a single process only and will not directly affect any other process. It may indirectly affect dependant processes but never contributing processes. For example, if a process were to feed an OR process and it had been acknowledged and all other inputs to the OR process were in the OK state then the output of the OR process would be in the acknowledge state.

Conversely, if a process feeding an OR process is in alarm state and the resulting alarm on the output of the OR process is acknowledged, then the original process will not be acknowledged as a result. This means where a summary alarm on a screen is acknowledged, any contributing alarms (which may show on another panel) will remain in their Alarm state.

### 2.3.2 AND Process

The output of an AND process will be the same as the lowest of the following levels of its inputs: -

- On/Force On (appears as Alarm)
- Latched
- Acknowledged
- Ignored
- Off/ Force off (appears as OK)

The inputs are configured in the same way as for the OR process.

### 2.3.3 NOT Process

The result of this type of process is the inverse of its input. The input is selected from a drop down list.

### 2.3.4 TRUTH Table Process

This process works as a logic truth table. The inputs are selected in the same way as for the OR and AND processes. The "inputs" are the truth table "rows". It also has a number of truth table "Columns" which indicate the required states of the inputs to make a TRUE (i.e. FAIL= Alarm) output. For example the TRUTH table in the image below has three inputs.

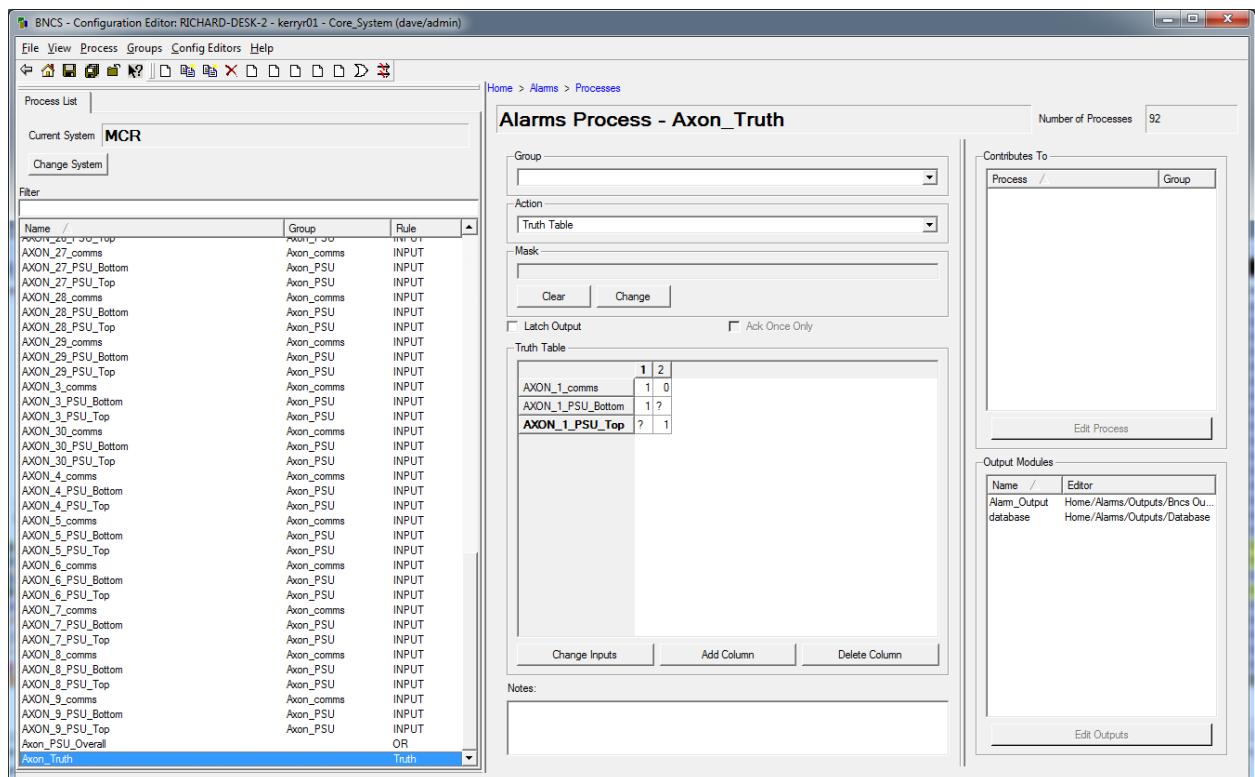
To add a column press the "Add Column" button.

Each entry must be one of either:

- 0      this input must be OK
- 1      this input must be Alarm
- ?      not interested in this input for this column.

Add more columns as necessary.

The output of the truth table will be TRUE (i.e. Alarm) if any of the columns are TRUE (it's an OR of the columns). In the example below the output will be true if either "Axon\_1\_PSU\_Bottom" AND "Axon\_1\_comms" are both in Alarm OR "Axon\_1\_PSU\_Top" is in Alarm AND "Axon\_1\_comms" is OK.



### 2.3.5 Holdoff

The output state becomes Alarm once the input has been in Alarm state for the designated number of seconds. Fractions of a second are allowed, to two decimal point resolution (ie hundredths of a second).

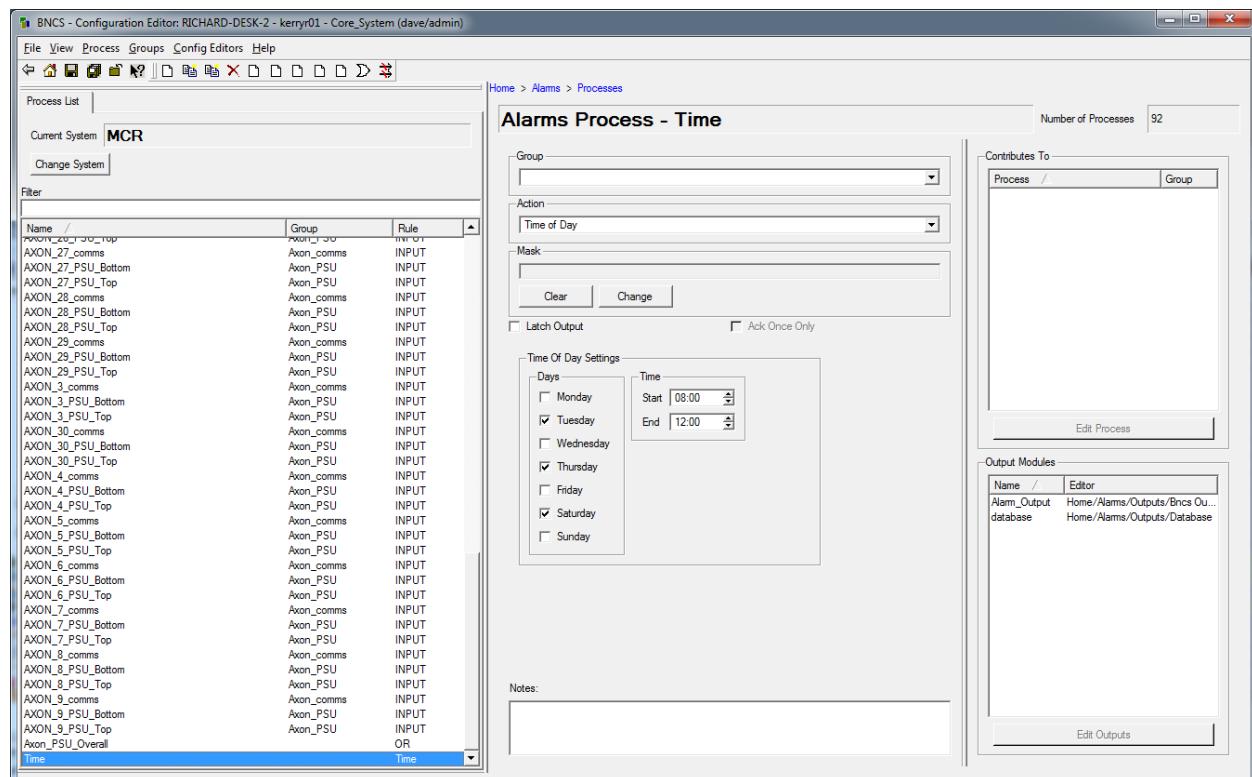
This may be used when you don't want to see a failure that lasted less than the given time period.

If possible it is suggested that the holdoff is done in the Acquisition module to avoid unnecessary processing.

### 2.3.6 Time

This has parameters for a set of days and start and end times.

If the day matches and the time is between the given start and end time the output will be TRUE (Alarm). For example the process below will be in the Alarm state every Tuesday, Thursday and Saturday between 0800 and 1200.



## 2.4 Properties Available for all Processes

Actions relating to processes may be done either by right clicking in the processes pane (ie using its context menu), via icons on the tool bar, or the Process menu item. Descriptions will not repeat this.

### 2.4.1 Group

This is a way of grouping alarms for ease of configuration. It has no effect on the logic or on the outputs.

New groups may be added using the Groups entry in the menu. From this groups may also be deleted or renamed. A group must exist before a process may be assigned to it.

The group associated with a highlighted selection of processes may be set or changed by using "Set Group".

### 2.4.2 Mask

It is possible to mask one process with another. Mask means that the process will not fail unless the masking process has failed too (similar to an AND process). This is often used when one process has to mask several others.

For example a timed function may be used to mask a process so that the alarm will only trigger at certain times of every day.

### 2.4.3 Latch

Latch means that a process will go from Ok to Alarm instantly but will not go from Alarm to OK unless it has been acknowledged. See the Alarm State diagram below.

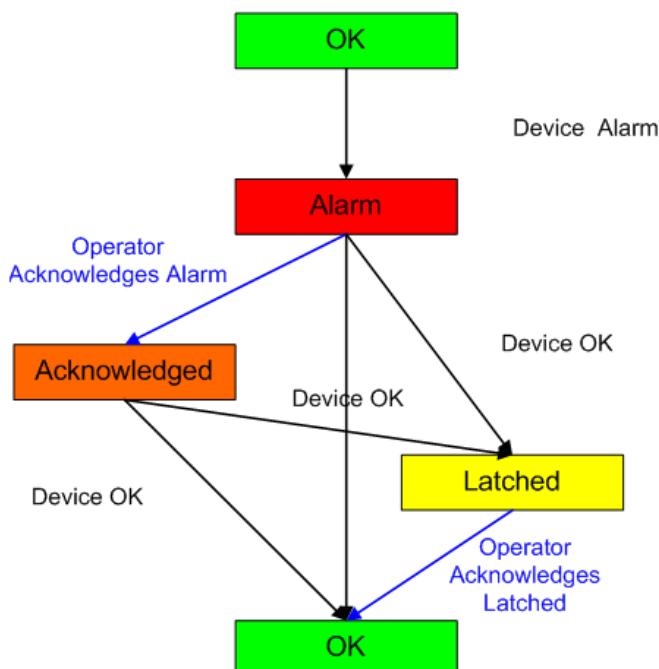
The setting may be changed using "Latch / Ack".

There are three possible options for the Alarm to OK transition.

If there is no latch then if the operator is able to acknowledge an alarm the steps will be Alarm -> Ack ->OK, the left hand path.

If the operator is not able/allowed to acknowledge the steps will be Alarm -> OK, the straight through path.

## Alarm State Cycle



### 2.4.3.1 Acknowledge Once Only

If the process is set to "latch" but not to "Ack once only" then the steps will be Alarm -> Acknowledge -> Latched -> OK, or Alarm -> Latched -> OK if the fault has cleared before the operator is able to acknowledge. It may be desirable for the operator only to have to acknowledge the process once. If this is the case then set the "Ack Once Only" box.

## 2.5 Duplicating Processes

### 2.5.1 Simple Duplication

One or more processes may be duplicated by selecting "Duplicate Proc". This will duplicate each process selected. The new processes will be named "copy\_n\_of\_processname", where "n" is a number and "processname" is the source process's name.

Any logic process (AND, OR etc.) copied will keep its original inputs.

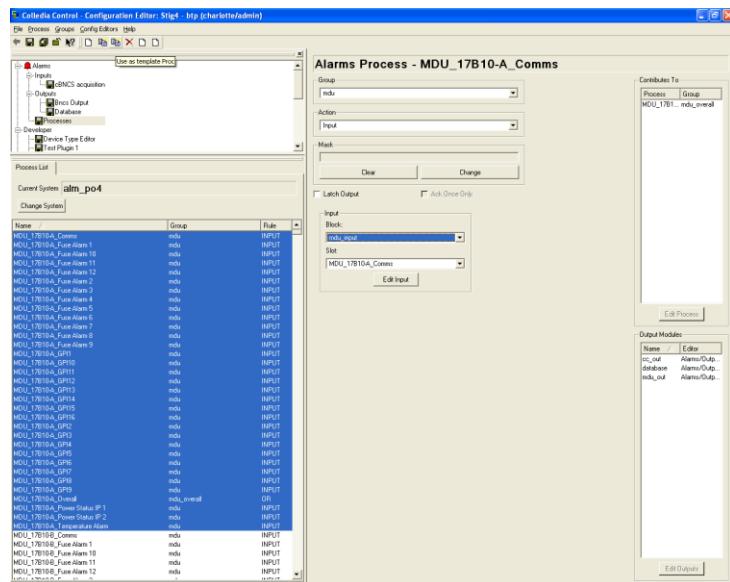
### 2.5.2 Template Duplication

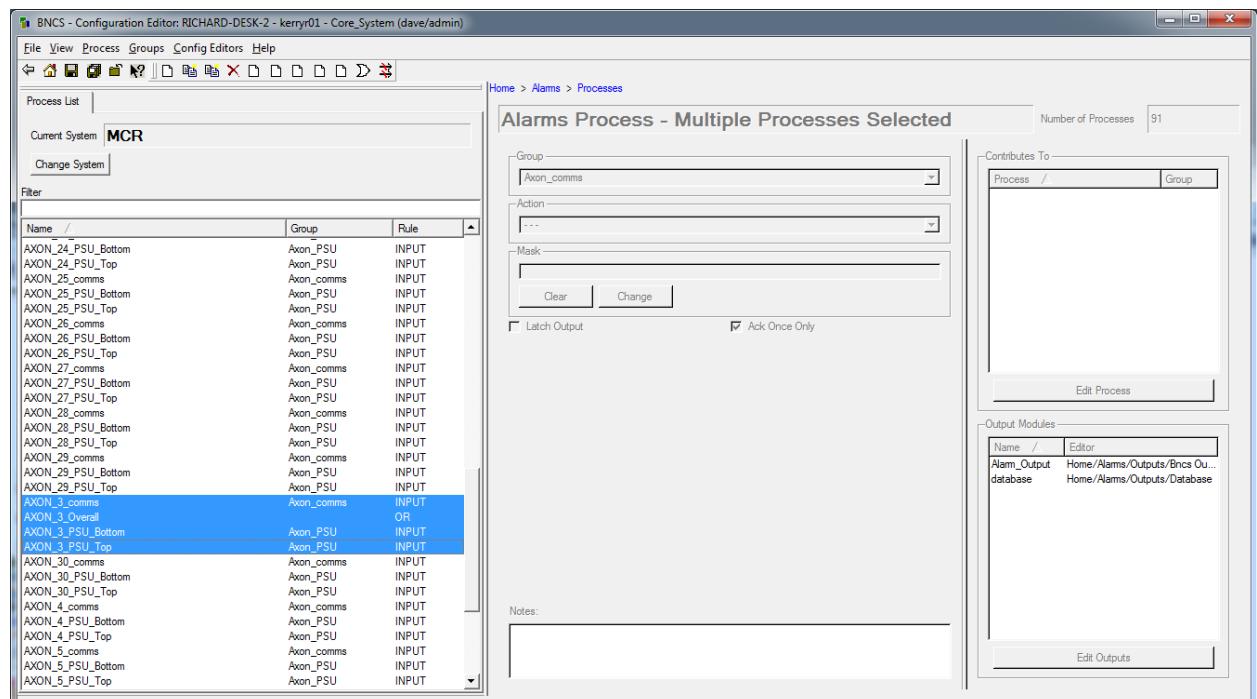
Duplication using a template may be used to duplicate a selection of processes with an option to maintain the relationships between them. This may be useful when there are groups of alarms that are repeated, for example for similar pieces of equipment or transmission chains.

Note that if you want to duplicate an OR process you must also select all its inputs before duplicating in order to maintain the relationships. When input processes are duplicated if there isn't a valid input with the new name the tool will create new processes and take the actual input from the original process.

What follows is an example of a duplication of all alarms from a device, where one alarm is an aggregate (OR process) of all of the others.

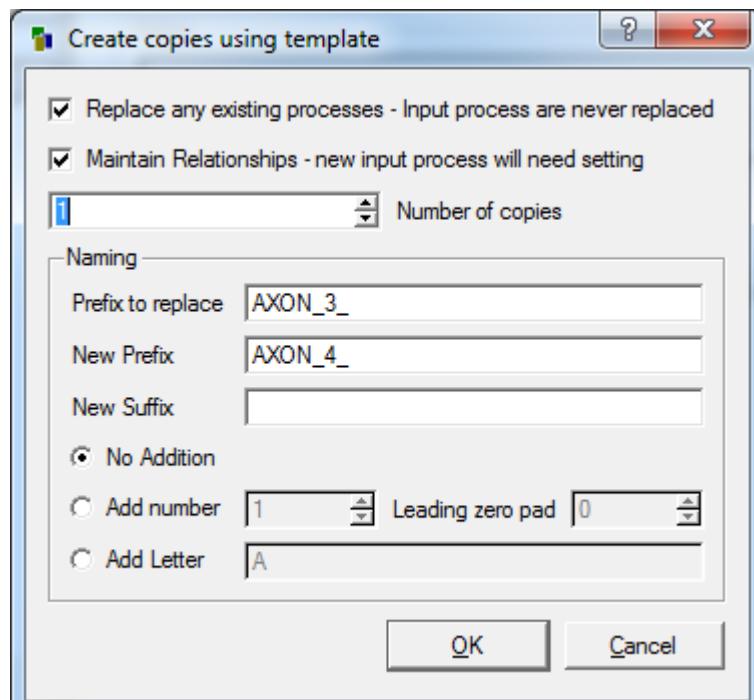
First select all the alarms that are to be used as a template, in this case it is all the alarms from AXON\_3. There are already input alarms from AXON\_4, but no OR process.





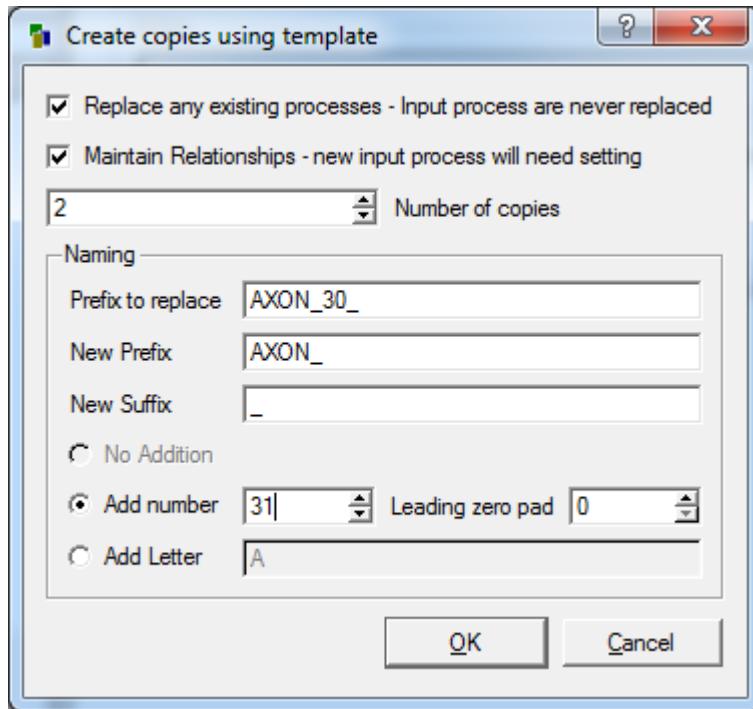
Then select "Copy Proc Using Template".

The tick boxes "Replace any existing processes" should be ticked, as should "Maintain relationships". Make sure the number of copies is 1. Replace the prefix "MDU\_17B10-A\_" with "MDU\_17B10-B\_". Press OK.



You should find a process "AXON\_4\_Overall" has been created and has the correct inputs and settings (Ack once only etc.). All the selected process have the AXON\_3\_ prefix removed and replaced with AXON\_4\_.

If making more than one copy it is necessary to select an option for how the new ones will be numbered. In that case both the new prefix and new suffix should be supplied.



The new processes' names are generated by concatenating the New Prefix value, the generated number and New Suffix value. In this example the two new processes will be numbered 31 and 32, as shown selected in the following image.

If the range to be generated would result in numbers of differing lengths they may be padded with leading zeroes to prevent sort order issues. The value supplied is the length the number strings will be extended to.

It is also possible to do multiple copies adding letters rather than numbers, for which use "Add Letter" and select the first.

## 3 Naming Convention

Careful selection of naming convention for processes makes sorting and duplicating inputs processes and outputs more straightforward. This is especially true for a large system with lots of similar devices or alarms.

It is recommended always to go from more general to more specific.

For example:

\*area\*\_\* chain/group of devices\*\_\*device\*\_\*parameter\*\_\*Alarm-parameter\*

It is often not necessary for all of these to be in the name, but the general principal should be followed.

## 4 Documents Referenced

This document should be read in conjunction with other documents in the tree.

In particular:

Alarm – overview

Configuration Server/Editor

alarm – mainapplication

File Format – Process Module

The documentation relating to file formats may also be useful.

## 5 Version history

### 5.1 Software Version

Version numbers shown here may not be seen within the software itself. The implementation date is a more reliable way of determining whether a particular issue is present in any particular instance of the software.

Version	Date	State / Changes	Author
1.00.00	15 November 2005	Original Release	Charlotte Bell
1.00.00	13 January 2004	Demo system	

2.00.00	31 March 2004	System now one application with plugin dlls	Charlotte Bell
4.5.0.2	19 <sup>th</sup> September 2007	Added the ignore intermediate states to the OR gate.	
4.5.1.0	21 <sup>st</sup> January 2008	Added the ability to filter the displayed processes.	

## 5.2 Document Version

Version	Date	State / Changes	Author
1.00.2	19 January 2015	This is now a separate document from the main app doc	Charlotte Bell
1.00.3		Reformatted to make more sense	
1.00.10	27 April 2009	Added section about duplicating processes	Pete Lasko
1.00.11		Added the ignore intermediate states to the OR gate.	
	26/01/2015	Document revised to current conventions and template. Corrections and amendments. Images reworked from current versions of software.	Richard Kerry

Atos IT Services Limited  
4 Triton Square  
Regent's Place  
London NW1 3HG, UK  
<http://uk.atos.net/>

BNCS  
Room G300  
Stadium House  
Wood Lane  
London W12 7TA  
[collediacontrol.it-solutions.gb@atos.net](mailto:collediacontrol.it-solutions.gb@atos.net)