BNCS Alarm System

Atos IT Services

BNCS Alarms and Monitoring System

Main Application

AlarmControl.exe

Written by: Charlotte Bell, Robert Bagyinszki, Richard Kerry

BNCS

Contents

BNCS

1 Alarms and Monitoring

1.1 Overview

This document describes the BNCS Alarm System. It is a configurable, scalable and modular application for monitoring diverse types of equipment.

Data is gathered from devices, pre-conditioned, manipulated by a selection of logic processes, and presented to the output modules. This enables current and historical status information to be viewed and logged.

1.2 Description

The BNCS Alarm System Main Application, AlarmControl, runs as an executable program on the BNCS network, along with a number of DLLs.

The Alarms System software consists of the host application, AlarmControl.exe, and input (acquisition) and output (presentation) modules, which are dynamic link libraries loaded by the host application.

One or more alarm systems may be run in one BNCS System. More than one alarm system may run on the same machine (subject to available CPU power). There may be one or more input and one or more output modules configured in any individual Alarm system.

The software requires configuration to perform any useful function. The key to using this software is setting up a configuration that fits your system and testing it with your system.

It is configured using configuration components in the BNCS V4.5 configuration client set.

It is strongly advised to design the system you require before starting attempting to configure anything.

The installation and configuration of the BNCS Alarms System requires an installation of the V4.5 BNCS System.

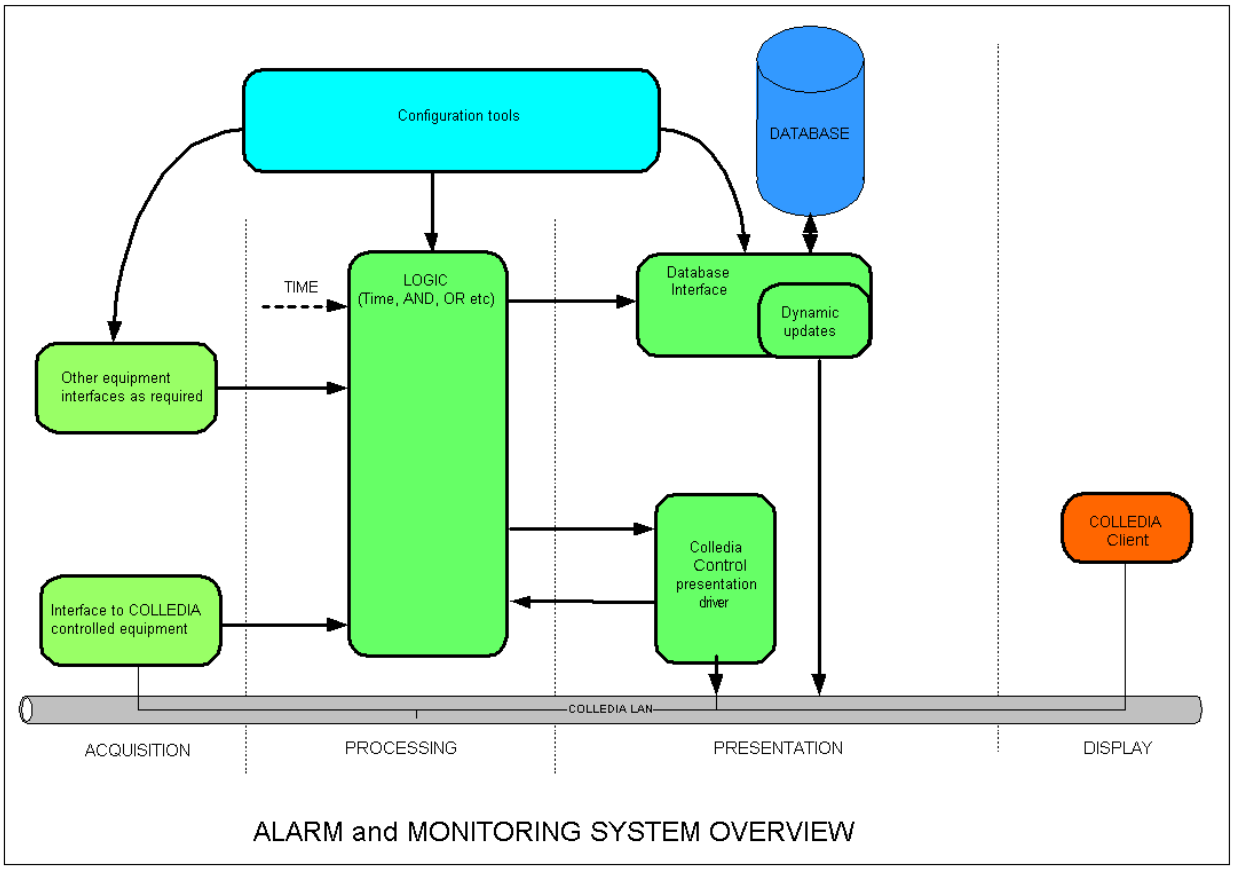
A typical system will have one or more input modules and one or more output modules configured.

At present two input modules are available:

1. The BNCS Acquisition Module, which runs as a BNCS client, gathering information from devices on the BNCS network.
2. The Driver Status acquisition module, which gets information from CSI regarding running drivers and their Tx/Rx status.

There are two output modules available:

1. The BNCS Output, which presents the outputs to the BNCS network. This supports both the reporting of alarm states to the BNCS network and also responds to data writes to override the states of the alarm processes.
2. The Database Interface Output which writes the to an SQL database. This writes the current state of processes to one table and appends to another table to form a historical log. It can also supply change information to a BNCS network and allow BNCS Clients to add information to the historical log.



ALARM and MONITORING SYSTEM OVERVIEW

The diagram above shows an overview of the system components, these are briefly explained below.

2 Acquisition / Inputs and Preconditioning

The BNCS Acquisition module gathers data from devices that have BNCS drivers e.g. GPI Crates, Satellite receivers, Glue components. It is also possible to gather information from devices that are only being monitored and not being controlled.

It is also possible to pass in alarm information from another alarm system.

The Driver Status acquisition module gathers data from CSI regarding whether drivers are running and their Tx/Rx status. Alarms may indicate whether a driver is running, or whether its resilience is as expected.

The incoming data is normalised to three standard states: OK, Alarm, or Unknown. Other processing e.g. timed hold-off can be applied at this point.

BNCS

3 Processing

All configured process values are calculated at runtime and made available to the output module(s) for reporting. Inputs, fixed time functions, holdoff plus logical operators (OR, AND, NOT, TRUTH) may be configured at configuration time. Changes to the configuration cannot be made at runtime.

Note: there is no inherent severity/priority for processes, they are all equal.

3.1 Native States

The 'native' states supplied by the Alarm System are: OK, Alarm, Latched and Unknown. The Unknown state is returned if the acquisition module has never had a revertive from the particular input.

Use of the latched state is optional and is configured for each process.

Value	Name	Description
0	Normal/OK	The process is not in a fault/alarm state.
1	On/Alarm	The process is in a fault/alarm state.
6	Latched	The process's input has gone to a fault/alarm state, and then back to Normal. No acknowledgement has been received.
-1	Unknown	The system has had no input regarding the process's current state.

3.2 Override States

A panel element (button) may be configured to override the native states. The possibilities are the following:

Value	Name	Description
2	Forced Off	Forces the process to appear OK.
3	Forced On	Forces the process to appear Alarm
4	Acknowledge	The operator has acknowledged the process
5	Ignored	One shot Force Off. Resets the next time the process goes to OK

Note: A process cannot be required to be OK. This will only happen when the input has returned to an OK state.

3.3 Override Commands

The following values will manipulate the state of the process (and sometimes others) but don't get stored with the process.

Value	Name	Description
7	Acknowledge Only	Ack if failed. Don't clear inhibits or Force on/off.
8	Clear	Clear inhibit. Clear Force on/off. Don't ack.
9	Acknowledge ACK_CONTRIBUTORS_IMMEDIATE	Acknowledges all the process's immediate contributors. Ie all those listed in the process's own configuration
10	Acknowledge BNCS_AL_ACK_CONTRIBUTORS_ALL	Acknowledges all the process's contributors, and all of theirs, recursively to the first level of input processes (ie those with no contributores).

3.4 Timed Overrides

The system is also able to apply an override for a specified period of time, after which it will automatically revert to Normal/Off.

Only the Bncs output module is able to access this functionality.

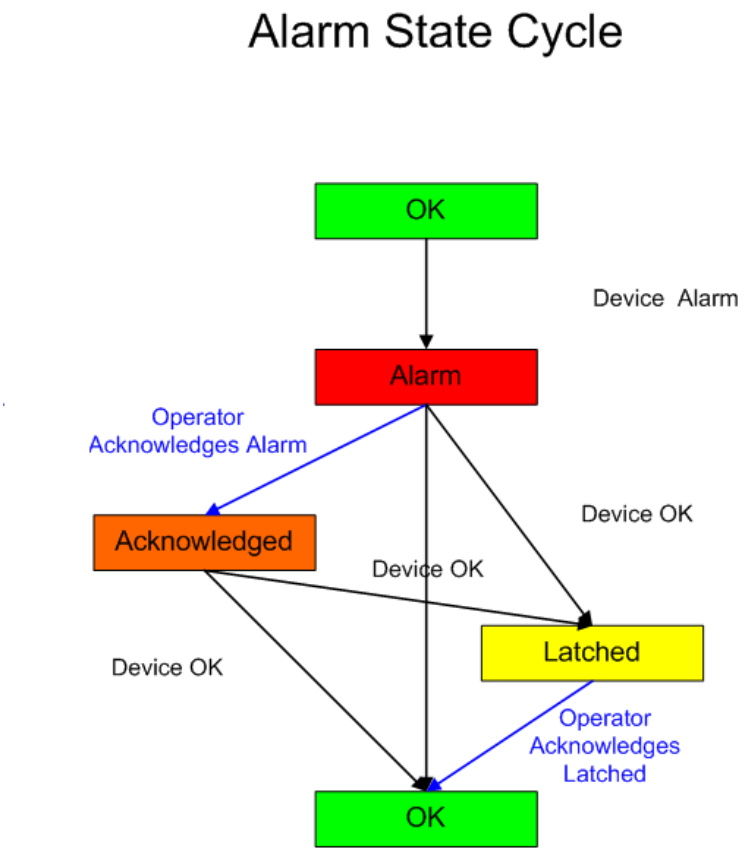
3.5 Alarm States

The diagram below shows the state cycle of a process. It shows the possible states that a process can be in as it goes from OK to Alarm to OK again.

If the process is not configured to latch then acknowledgement is optional.

If it is configured to latch then it may be configured to only require one acknowledgement per alarm-OK transition.

This configuration is done at configuration time.



4 Output/Presentation

At present there are two output modules available, for BNCS Output and Database output.

4.1 BNCS Output

This module will present the states of configured processes as devices on the BNCS network (Infodriver presentation).

Panels interact with these devices to show the status of the processes and to perform any overrides.

4.2 Database Output

This module will present process information to a SQL database on the network to store the current states of configured processes and as a historical log.

It logs the date/time, process name, device name, state, group, severity (5 =major alarm to 1= information).

It also broadcasts changes to the BNCS network to enable appropriate clients to display a view of the information.

When an alarm is forced off, this will be logged, but any changes to the actual source will not be logged until the override is cleared.

5 BNCS Client

This is a normal BNCS client and is not part of the alarms system itself. It could be an alarm panel, or a mixture of a control, navigation and alarms functionality.

If alarms are presented to the client it is possible for the operator to override the state. For example, the operator could acknowledge an alarm in the Alarm state or Force On an intermittent alarm while the fault is being dealt with by an engineer.

A BNCS V2 client can display the current states and action overrides via the BNCS Output module.

In order to display the log control the BNCS V4.5 client is required (or any 2 bit client with SQL support).

6 Configuration procedure

Each module must be configured for the system to do anything useful.

This is done using the BNCS Configuration editor and server.

The configuration editor allows new alarms and systems to be added, and configuration of the inputs, processes and the outputs.

See configuration documentation for details.

7 Starting up

The application AlarmControl.exe is run with the argument "/AlarmSystem=name", where "name" is the alarm system name. AlarmControl.exe will load the process configuration and the appropriate input and output modules on start-up.

See also the command-line options section below.

For the BNCS Acquisition Module to work CSI.exe must be running in client mode. For the BNCS Output module to work CSI.exe must be running in driver mode and the appropriate infodriver (i.e. the one set in the output configuration) must be running prior to AlarmControl.exe.

AlarmControl is typically run from the BNCS Workstation Manager. To do this the required elements must be included in the Launch configuration.

8 Resilience

AlarmControl is able to be run on two separate workstations as a resilient pair for any particular alarm system. The instances send messages over the network via UDP. The

messages enable the two instances to agree which is able to transmit (ie it is considered Tx/Rx) and which is not (ie it is considered RxOnly).

If resilience is not needed, or for development purposes, the generation and use of these messages may be disabled by running with the command-line argument "/UDP=no".

9 Diagnostics

The system shown below has one of each type of plug-in:

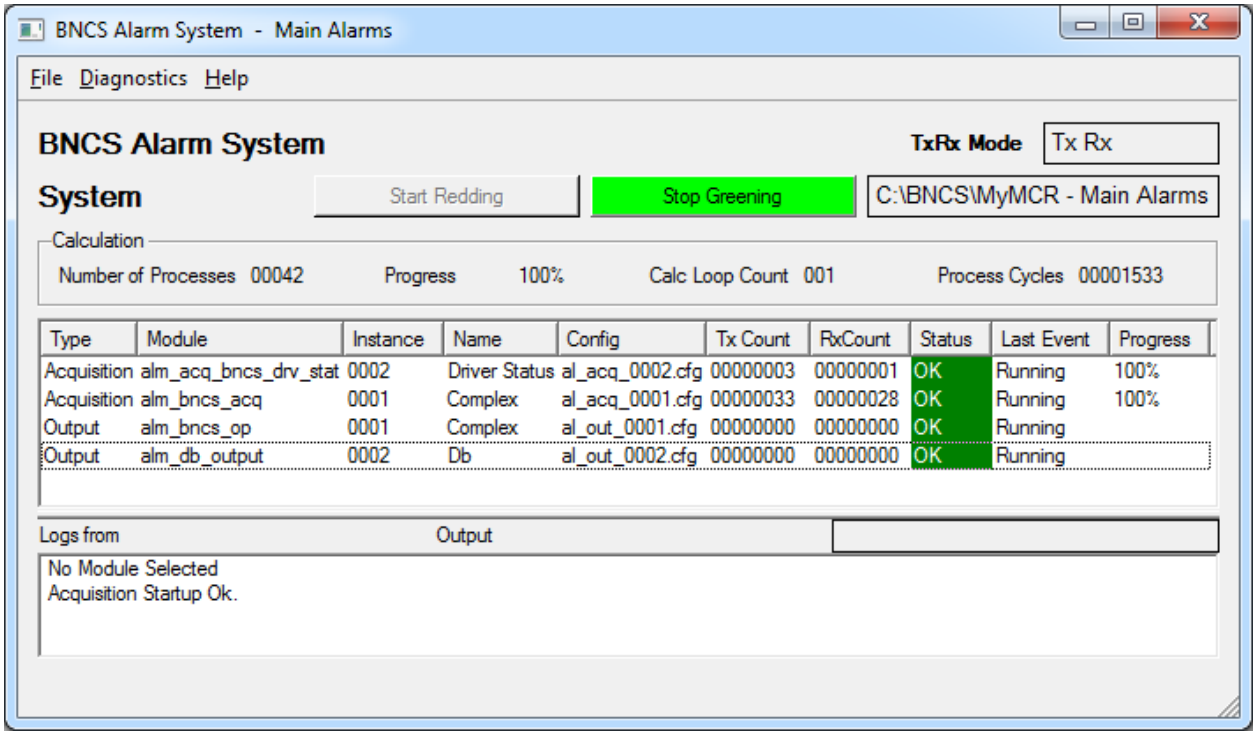
BNCS Acquisition

BNCS Driver Status Acquisition

BNCS Output

Database Output

The path is "C:\BNCS\MyMCR" and the alarm system name is "Main Alarms".



The display is updated approximately once per second.

The system name is displayed at the top of the dialog along with the Tx/Rx status.

The "Calculation" box contains:

The number of processes in this configuration.

The progress through the initial polling process, expressed as the percentage of inputs successfully polled.

The "Calc Loop Count" which is the number of times the program has run through the calculations in order to calculate all the processes. If this value is 200 there is a serious

fault with the configuration that needs to be fixed (it is unable to calculate one or more processes possibly because of a logical tautology or a non-existent input).

The "Process Cycles" which is the number of process cycles completed since the program was started.

The list below this displays all the input and output modules and various properties:

Type – Acquisition or Output

Module type – The name of the module DLL.

Instance – The number of the input or output config file.

Name – The name configured.

configuration file name – The name of the input or output config file.

Tx and Rx counts – Counts of BNCS messages sent or received.

Status – Whether the module is running properly.

Last Event – A brief note indicating the modules current phase of activity.

Progress – The progress through the initial polling process, expressed as the percentage of inputs successfully polled.

Any error messages for a module will appear in the bottom display when you select the module in the list.

For the BNCS Acquisition module, the message "All valid inputs received" will be displayed once a revertive has been received from each configured input.

9.1 Diagnostics Menu

In the main menu there is a "Diagnostics" entry. This provides options to control various settings and details of the application's behaviour, and can display a dialog showing all the processes with their current values.

9.2 Diagnostics Dialog

Select Diagnostics|Show to activate the diagnostic view of all the processes. This shows the state of all the processes and allows us to override their states. This may be used for testing and to track down errors in configuration.

The screen is shown below.

For each process the name, rule and state (both number and name) are shown, and any mask. Depending on their state the processes have different colours which are the same colour as the default colours for alarm controls on device panels.

9.2.1 Filters – Name and Inputs

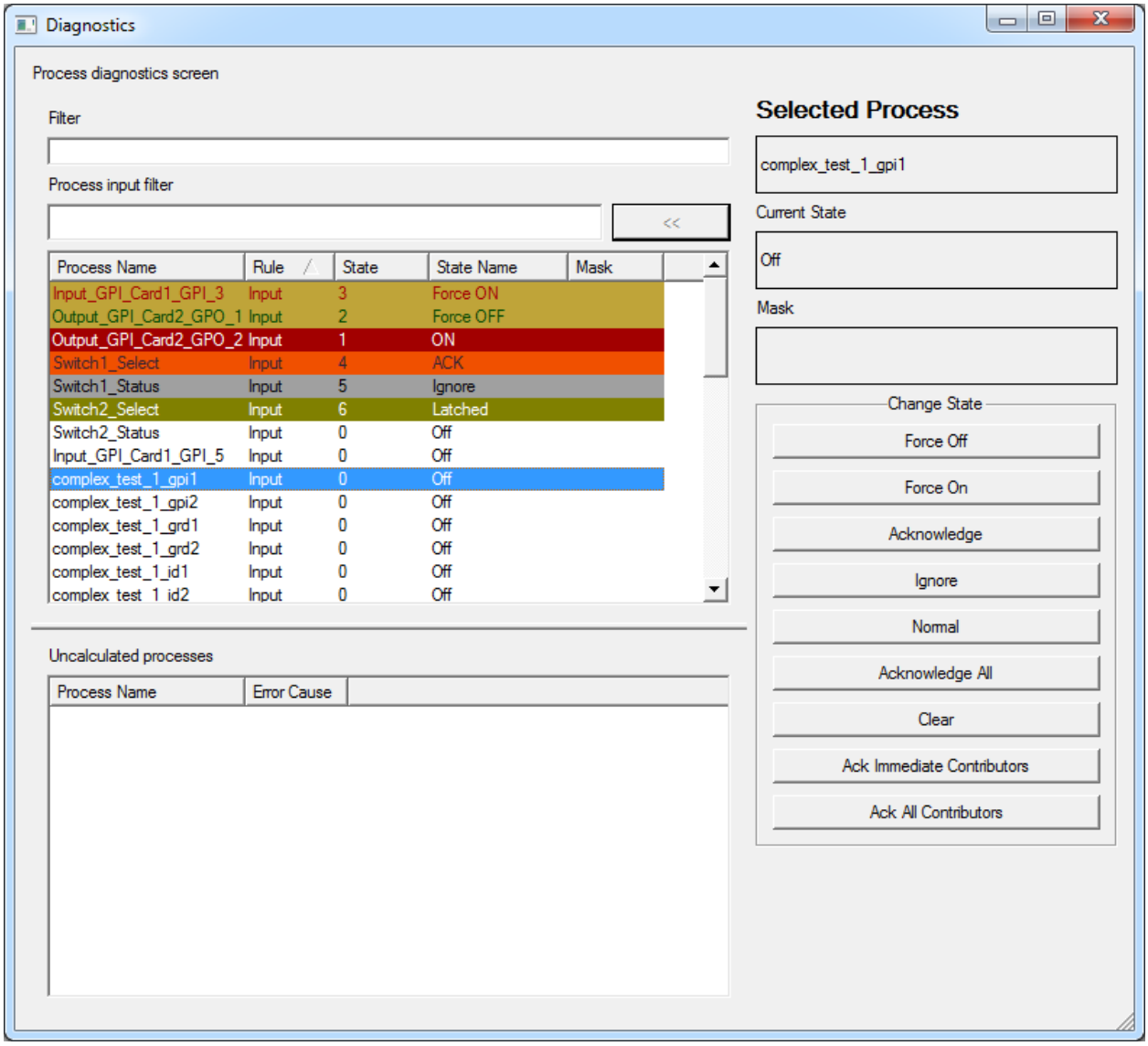
There are options to filter the process display by process names, or to display a process's inputs processes.

Entering text in the "Filter" box will do a simple wildcard match on the process names. Only processes matching the text will be displayed. This is case-sensitive.

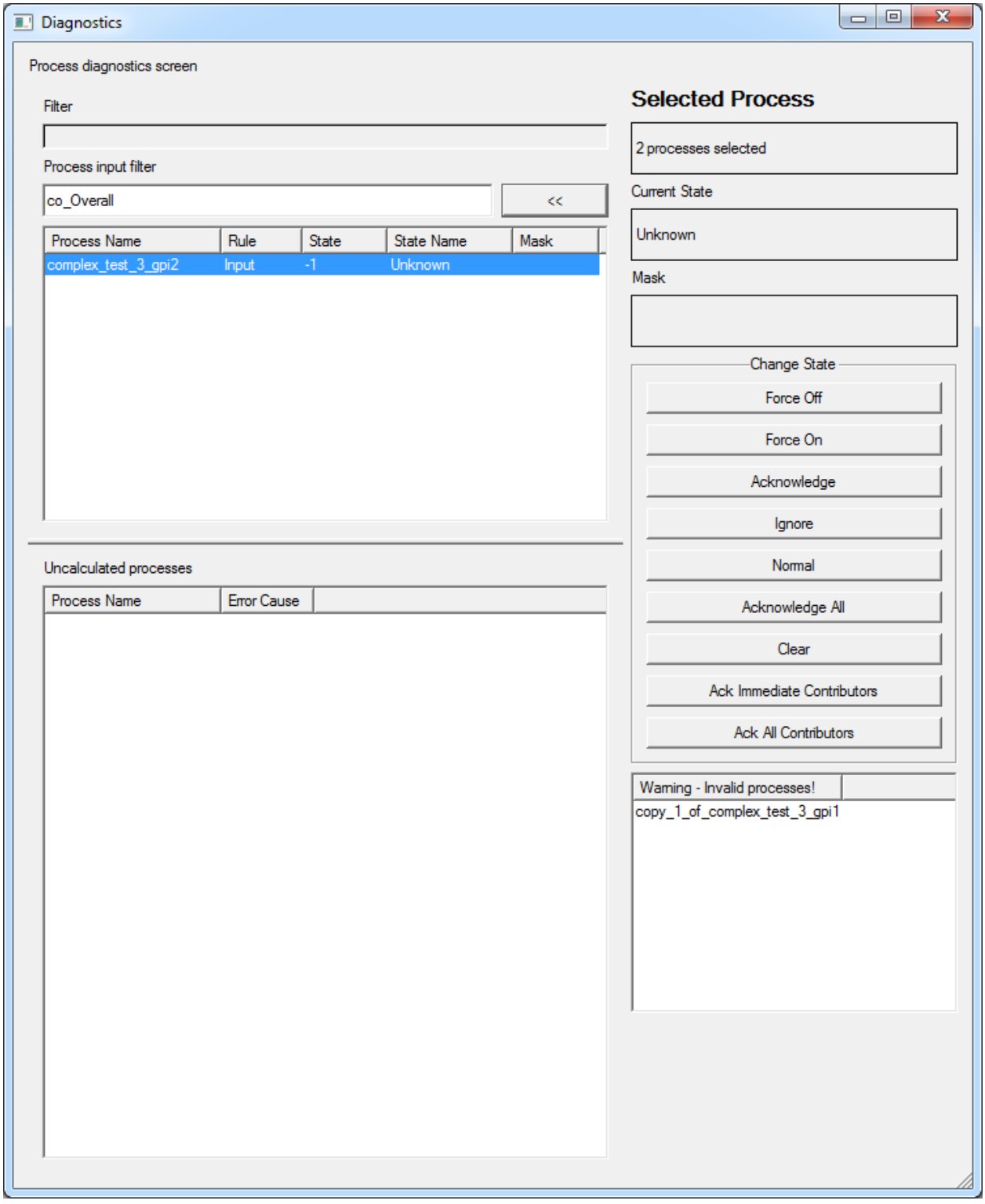
Entering the name of a process into the Process input filter will cause only processes contributing to that process to be displayed. This requires an exact match to the process name. The input filter can be filled automatically by double clicking on an item from the process list.

The process name filter will be disabled if there is anything typed into the input filter edit box. It may be activated by clearing the input filter.

The input filter is to assist in detecting the cause of the current process state. E.g. it can be checked quickly from the input list of an "OR" process where an "ON" state is coming from.



The input filter may also be used for detecting configuration errors in the process input list. Any invalid processes (ie non-existant process names) will be shown in the list at the bottom right of the dialog.


Diagnostics
Process diagnostics screen
Filter
Process input filter
co_Overall
<<
Process Name
Rule
State
State Name
Mask
complex_test_3_gpi2
Input
-1
Unknown
Uncalculated processes
Process Name
Error Cause
Selected Process
2 processes selected
Current State
Unknown
Mask
Change State
Force Off
Force On
Acknowledge
Ignore
Normal
Acknowledge All
Clear
Ack Immediate Contributors
Ack All Contributors
Warning - Invalid processes!
copy_1_of_complex_test_3_gpi1

9.3 Extension for Testing – "Greening"

The "greening" system can be used for setting an entire alarm system to normal state by correcting all the failed devices, ie by writing OK values to the various drivers.

It is only useful for testing purposes. **It is very important that it is not used on a live system** as it will write values to the drivers which are likely to be different from their "correct" values.

9.3.1 "Redding"

It is also possible to automatically write values to the drivers to put all the processes into an Alarm state. This is known as "Redding". It operates in a similar way to the Greening system.

The two similar systems – Greening and Redding – are together sometimes termed "Colouring".

9.3.2 Operation

The Greening system may be configured to start automatically at start-up, or may be activated manually. If it is set to start automatically the process will be started and will run either for the specified time, or will be stopped automatically when all the inputs are reporting OK.

In action when any input process reports a fail state the system selects a value which will cause the process to respond as OK and sends it to the relevant driver. For Discrete processes a value corresponding to OK will be chosen from the list of states available for the control. For Analogue inputs a value is calculated based on the configured settings.

Redding works similarly but values are chosen in the opposite sense – to generate fault values.

If colouring is enabled but not started automatically it may be started using one of the buttons. In that case the scanning process will be restarted repeatedly at the period configured (by AutomaticGreeningTimeout or AutomaticReddingTimeout). Scanning will be repeated until turned off by the button.

9.3.3 Configuration – Greening.ini

The system is controlled by entries in the greening.ini file, which is in the root directory of the alarm system (ie <System root>\config\alarms\<Alarm System>\greening.ini).

If the file doesn't exist it will be generated automatically during startup.

If any of the properties are not present the defaults will be written.

The following parameters may be set in greening.ini.

9.3.3.1 [Colouring]

This section contains parameters affecting both parts of the "colouring" system.

Parameter	Default	Details

Parameter	Default	Details
EnableSetInitialState	false	If colouring is activated by setting this parameter to true, two buttons will appear on the alarm system GUI for manually starting or stopping colouring. If colouring is activated automatically at start-up (see below) the corresponding button will be coloured.

9.3.3.2 [Greening]

This section contains parameters affecting only "greening" – ie the automatic writing of values selected to cause the alarm processes all to report as "OK".

Parameter	Default	Details
SetInitialStateOK	false	Selects whether greening is automatically activated at start-up. setting the control and alarm instances automatically to state 'OK' after startup
SetInitialStateOKTimeout	300 (seconds)	Greening will be stopped after the number of seconds configured here. If it couldn't be finished the remaining devices keep their states. Updated devices and devices which couldn't be updated because of the timeout will appear in the log of the input file. This setting relates to "greening" when started automatically on program startup.
AutomaticGreeningTimeout	30 (seconds)	If greening is running, activated using the button, the colouring process will be restarted every AutomaticGreeningTimeout seconds. This setting relates to "greening" only when started by the Start Greening button.

9.3.3.3 [Redding]

This section contains parameters affecting only "redding" – ie the automatic writing of values selected to cause the alarm processes all to report as "Alarm".

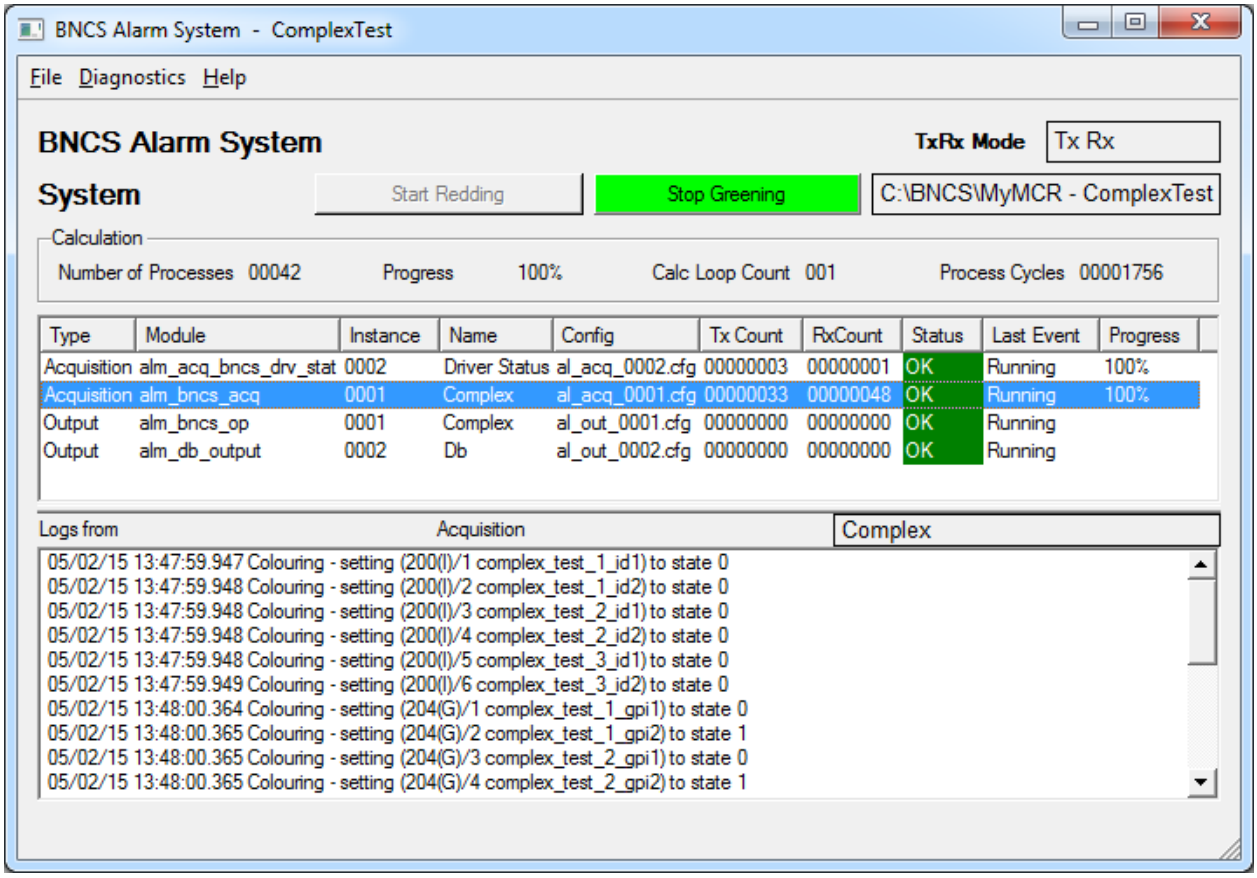
Parameter	Default	Details

Parameter	Default	Details
SetInitialStateFail	false	Selects whether redding is automatically activated at start-up.
SetInitialStateFailTimeout	300 (seconds)	Redding will be stopped after the number of seconds configured here. If it couldn't be finished the remaining devices keep their states. Updated devices and devices which couldn't be updated because of the timeout will appear in the log of the input file. This setting relates to "redding" " when started automatically on program startup.
AutomaticReddingTimeout	30 (seconds)	If redding is running, activated using the button, the colouring process will be restarted every AutomaticReddingTimeout seconds. This setting relates to " redding" only when started by the Start Redding button.

Greening can only be started manually if the alarm control application is in the Tx/Rx state.

If greening is active, use of the "LATCHED" state is disabled from alarm panels or any other client requiring to manipulate process states.

If the system has finished polling all the inputs the greening buttons may be visible but are disabled and may not be clicked.



9.3.4 Dependencies

AlarmControl.exe requires the QT dll (currently qt-mt323.dll), bncsif.dll and the Visual C runtime dll MSVCP60.dll.

It also requires Calarm_process.dll which does all the process calculations.

10 Command-Line Options

There are a number of command-line options for this program.

10.1 /AlarmSystem

This must always be provided.

Syntax is : /AlarmSystem='name', where 'name' is the name of the alarm system. Ie the folder name within the config folder where the alarm configuration is found.

10.2 /Period

(Optional)

Syntax is : /Period=20

This is the time interval for running the alarm process calculations. Unit is milliseconds. Default value is 20(ms).

The alarm system will wait for a short period after startup and then will attempt to perform all the process calculations repeatedly at this interval. This can be a drain on CPU resources and may be reduced by using this parameter.

10.3 /Log

(Optional)

This controls logging from certain parts of the program.

Syntax is : /Log='number'. The number should be in the range 0 to 63 and are interpreted as a six bit number. The six bits control six diagnostic flags as follows.

1: AlarmControl itself.

2: The acquisition and output Plug-ins

4: UDP. Ie the mechanism used to ascertain bus mastership.

8: Mastership. Transitions of bus mastership.

16: Per-Session Logging. Each run of the program will create a folder, named according to the start time, in which any log files are written.

32: Verbose Logging.

In all cases these values are passed into the acquisition/output modules, which are able to interpret them as they wish. The meanings above are the general uses but individual modules may differ.

10.4 /ASFU

(Optional)

This stands for "Acquisition Startup Failure Unrecoverable".

The program usually assumes that any problem in any plug-in can be recovered from and that the plug-in will do so as soon as it can. In some circumstances, with some plug-ins, this is not possible. In particular it is sometimes found that a problem reported by the Acquisition plug-in is an indication of a fundamental problem from which the plug-in (and the main program) is not able to recover. In this case it is necessary that the whole program should be shut down.

To activate this behaviour use this option.

10.5 /NetDelay

(Optional)

Syntax is : /NetDelay='number'.

Some output plug-ins have the facility to start-up with a reduction to the amount of network traffic (ie BNCS traffic) they produce. A certain time after start-up these will be requested to return to the normal amount of traffic. The purpose is to reduce the amount of unnecessary information being sent to the network during the start-up of the alarm system, which always produces a large amount of traffic during its start-up.

This time delay, in milliseconds, is calculated by multiplying the parameter by the number of processes in the system. The default is 60ms per process.

At the moment this parameter is only used by the database output module.

The following qualifiers are not being documented as they are for development/diagnostic use only : udp, localconfig, cc_root, cc_system, eventdriven.

11 Logging

When Logging is enabled (via AlarmControl's GUI or arguments), selected messages are logged into a file at a path determined by the BNCS settings.

Log files are written in

`%CC_ROOT%\%CC_SYSTEM%\logs\alarms\<name>`,

where <name> is the alarm system name.

If per-session logging is enabled, the log files are written to

`%CC_ROOT%\%CC_SYSTEM%\logs\alarms\<name>\<date-time>`

where <date-time> is the date and time at which the application started.

12 Documents referenced

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

In particular:

Various Acquisition and Output module documents.

Various documents on Acquisition and Output module configuration.

The documentation relating to file formats may also be useful.

13 Version history

13.1 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
2.00.00	31 March 2004	System now one application with plugin dlls	Charlotte Bell

4.5.7.283	21 November 2008	After CCM Additions.	Robert Bagyinszki

13.2 Document Version

Version	Date	State / Changes	Author
1.00.00	31 March 2004	Original Release	Charlotte Bell
1.1		Processing configuration details	Charlotte Bell
2.00	17 November 2005	Reworking	Charlotte Bell
2.1	30 November 2005	Reworking to fill out bulleted sections	Charlotte Bell
2.2	27 April 2009	Updated formatting to bring in line with other existing docs	Pete Lasko
	02/02/2015	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
colleadiacontrol.it-solutions.gb@atos.net