Newflow

NÅNO

Visual C||CURE
Manual

Newflow Ltd
George House
Derwent Road
Malton, North Yorkshire
YO17 6YB
UK

Tel:	+44 1653 310 000
Fax:	+44 8700 667 325
Email:	sales@newflow.co.uk

Document Information

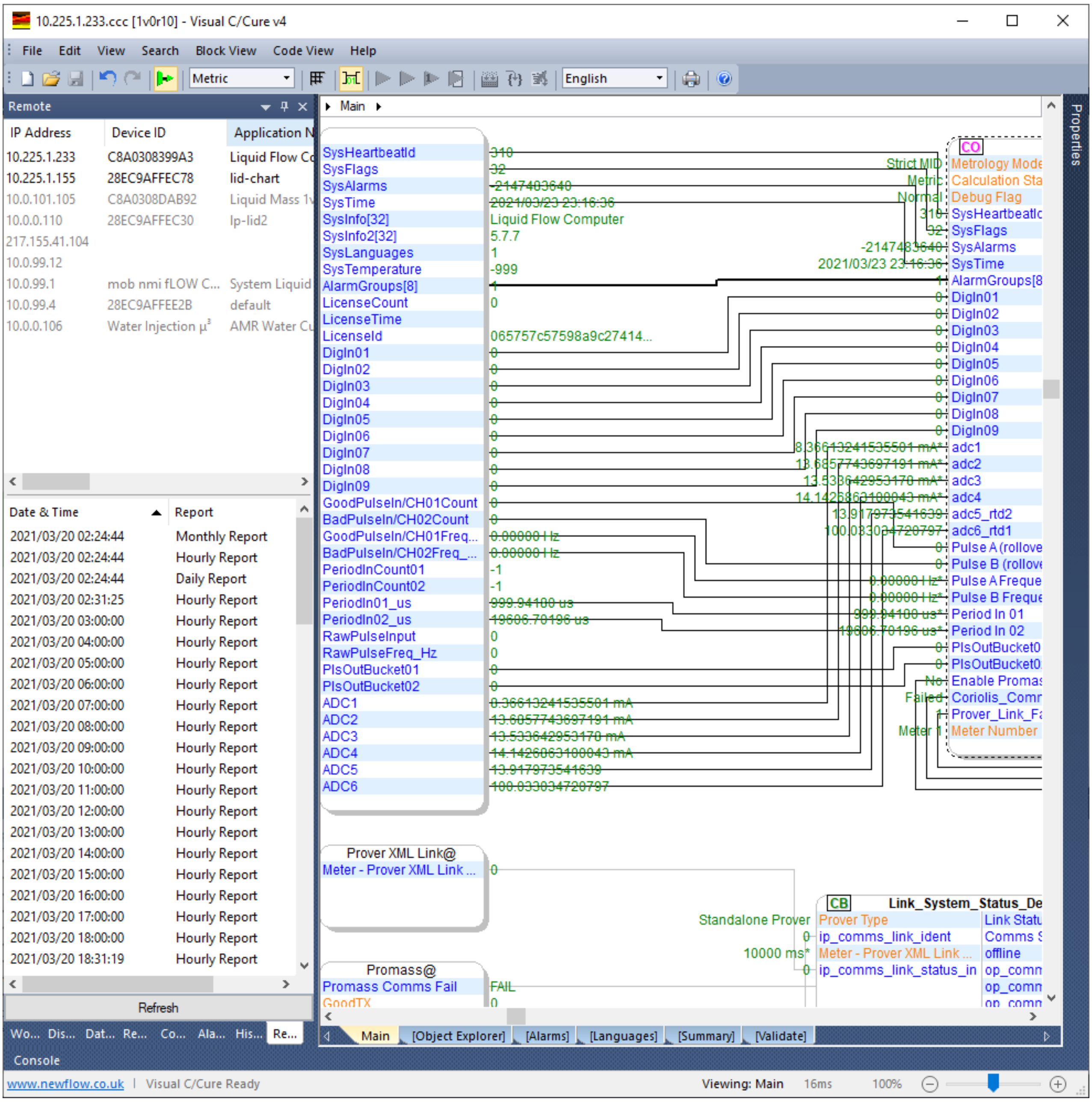
Document Name:	NÅNO Visual C||Cure Manual
Document Identifier	NFVCM/3

Rev	Date	Changes	Prepared	Checked	Authorised
0	25/06/13	1st Iteration	MPFJ	MOB	GPL
1	13/11/13	Includes 3V3 updates	MPFJ	MOB	GPL
2	29/04/22	v4 updates (to r8948)	MPFJ	GPL	MPFJ
3	07/02/23	v4 updates (to r9169) - Add ‘p’ display item style - Codeblock variables bar added - Add Datamaps section - Update Validate section - Add isinf() function	MPFJ	GPL	MPFJ

This document, submitted in confidence, contains proprietary information and shall not be reproduced or transferred to others without prior written permission of Newflow Ltd.

This document shall not form part of any contract. Specifications are subject to change without notice and Newflow Ltd accepts no liability of any kind for errors or omissions.

Full contractual terms are available on the website at www.newflow.co.uk/pdf/tandc.pdf



IP Address	Device ID	Application N
10.225.1.233	C8A0308399A3	Liquid Flow Co
10.225.1.155	28EC9AFFEC78	lid-chart
10.0.101.105	C8A0308DAB92	Liquid Mass 1v
10.0.0.110	28EC9AFFEC30	lp-lid2
217.155.41.104		
10.0.99.12		
10.0.99.1	mob nmi fLOW C...	System Liquid
10.0.99.4	28EC9AFFEE2B	default
10.0.0.106	Water Injection µ³	AMR Water Cu

Date & Time	Report
2021/03/20 02:24:44	Monthly Report
2021/03/20 02:24:44	Hourly Report
2021/03/20 02:24:44	Daily Report
2021/03/20 02:31:25	Hourly Report
2021/03/20 03:00:00	Hourly Report
2021/03/20 04:00:00	Hourly Report
2021/03/20 05:00:00	Hourly Report
2021/03/20 06:00:00	Hourly Report
2021/03/20 07:00:00	Hourly Report
2021/03/20 08:00:00	Hourly Report
2021/03/20 09:00:00	Hourly Report
2021/03/20 10:00:00	Hourly Report
2021/03/20 11:00:00	Hourly Report
2021/03/20 12:00:00	Hourly Report
2021/03/20 13:00:00	Hourly Report
2021/03/20 14:00:00	Hourly Report
2021/03/20 15:00:00	Hourly Report
2021/03/20 16:00:00	Hourly Report
2021/03/20 17:00:00	Hourly Report
2021/03/20 18:00:00	Hourly Report
2021/03/20 18:31:19	Hourly Report

SysHeartbeatId	310
SysFlags	32
SysAlarms	-2147483648
SysTime	2021/03/23 23:16:36
SysInfo[32]	Liquid Flow Computer
SysInfo2[32]	5.7.7
SysLanguages	1
SysTemperature	-999
AlarmGroups[8]	1
LicenseCount	0
LicenseTime	
LicenseId	065757c57598a9c27414...
DigIn01	0
DigIn02	0
DigIn03	0
DigIn04	0
DigIn05	0
DigIn06	0
DigIn07	0
DigIn08	0
DigIn09	0
GoodPulseIn/CH01Count	0
BadPulseIn/CH02Count	0
GoodPulseIn/CH01Freq...	0.00000 Hz
BadPulseIn/CH02Freq_...	0.00000 Hz
PeriodInCount01	-1
PeriodInCount02	-1
PeriodIn01_us	999.94100 us
PeriodIn02_us	19606.70196 us
RawPulseInput	0
RawPulseFreq_Hz	0
PlsOutBucket01	0
PlsOutBucket02	0
ADC1	8.36613241535501 mA
ADC2	13.6057743697191 mA
ADC3	13.533642953170 mA
ADC4	14.1426863100043 mA
ADC5	13.917973541639
ADC6	100.033034720797

CO

Strict MID	Metrology Mode
Metric	Calculation Sta
Normal	Debug Flag
310	SysHeartbeatId
32	SysFlags
-2147483648	SysAlarms
2021/03/23 23:16:36	SysTime
1	AlarmGroups[8
0	DigIn01
0	DigIn02
0	DigIn03
0	DigIn04
0	DigIn05
0	DigIn06
0	DigIn07
0	DigIn08
0	DigIn09
8.36613241535501 mA*	adc1
13.6057743697191 mA*	adc2
13.533642953170 mA*	adc3
14.1426863100043 mA*	adc4
13.917973541639	adc5_rtd2
100.033034720797	adc6_rtd1
0	Pulse A (rollove
0	Pulse B (rollove
0.00000 Hz*	Pulse A Freque
0.00000 Hz*	Pulse B Freque
999.94100 us*	Period In 01
19606.70196 us*	Period In 02
0	PlsOutBucket0
0	PlsOutBucket0
No	Enable Promas
Failed	Coriolis_Comr
1	Prover_Link_F
Meter 1	Meter Number

Prover XML Link@

Meter - Prover XML Link ...	0

Promass@

Promass Comms Fail	FAIL
GoodTX	0

CB Link_System_Status_De

Standalone Prover	Prover Type	Link Statu
0	ip_comms_link_ident	Comms S
10000 ms*	Meter - Prover XML Link ...	offline
0	ip_comms_link_status_in	op_comm
		op_comm
		op_comm

www.newflow.co.uk | Visual C/Cure Ready   Viewing: Main   16ms   100%

Table of Contents

1 Brief Description

Visual C||Cure is a Rapid Application Development (RAD) package, allowing software applications to be designed and written for the target hardware using an intuitive graphical environment.

Applications can be written and tested on a PC platform, without the need for any target hardware.

Remote targets can be fully managed, allowing application and firmware updates to be sent over the local network or Internet.

Live debugging of remote targets allows for a very powerful development system, helping developers and support engineers to support new and existing target installations.

Other features includes:-

- report designer
- alarms
- display configuration
- data charts
- field communications configuration

2 Installing onto a PC

To install Visual C||Cure onto a PC, run the .MSI installer and follow the prompts.

Each copy of Visual C||Cure requires a license in order to use.

The first time the software is run, it will display an About box, showing a serial number and blank registration key. Contact your supplier with your serial number and they will provide you with a registration key, which should be entered into the relevant box.

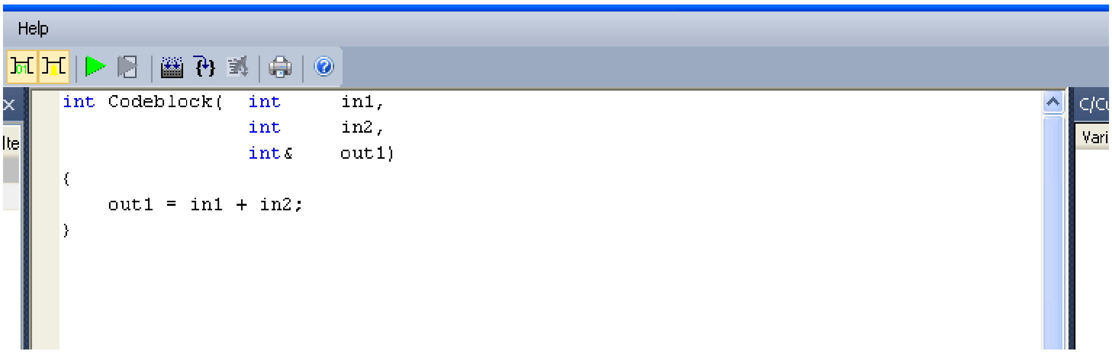
Once a valid key has been entered, the software will load.

3 How to Write An Application

All C||Cure applications are written using the Visual C||Cure IDE. This is a fully featured editor that allows code to be entered and data flow to be defined using a graphical environment.

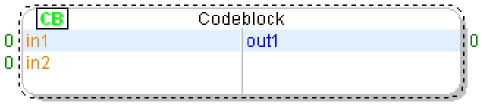
Applications are based on hierarchical containers (which can hold other containers and/or codeblocks) and codeblocks (which can only contain C code, as described in 7 - C||Cure Syntax).

To create an application, C code is written into one or more codeblocks.

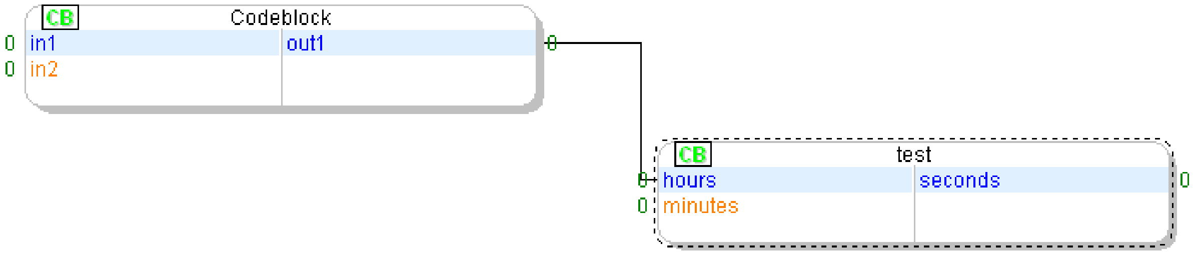


```
int Codeblock(  int      in1,
                int      in2,
                int&     out1)
{
    out1 = in1 + in2;
}
```

These codeblocks are then shown as graphical blocks within one (or more) containers, with codeblock inputs and outputs converted to “pins”.



Outputs from codeblocks and containers can then be “wired” to the inputs of other codeblocks and containers denoting the “flow of data” through the application.

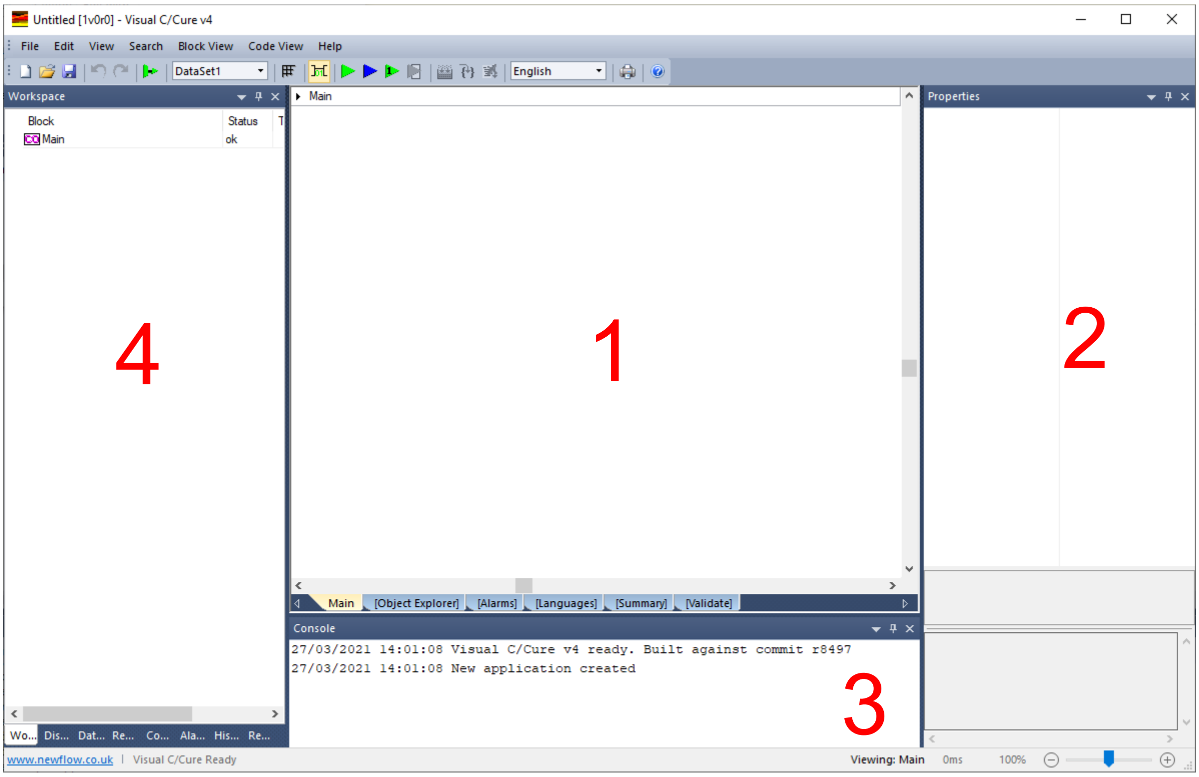


The pin colour denotes whether the data is persistent (orange) or not (blue). See 6.13 - Wrapped / Persistent / Unconnected Pins for more information.

Note: to ensure predictability, you can only connect input and output pins of the same data type and array size.

4 Editor Environment

The default editor layout shows 4 main windows:-



```
27/03/2021 14:01:08 Visual C/Cure v4 ready. Built against commit r8497
27/03/2021 14:01:08 New application created
```

1) Main editing window, where containers and codeblocks can be created, edited, moved, deleted and “wired” together.
2) Properties window, where the parameters of input and output pins can be viewed or modified.
3) Console window, where messages generated by the editor are displayed.
4) Information section, where other information windows are housed. These include Workspace, Displays, Comms, Datamaps, Reports/Live Screens, Alarms, Remote Targets and Historical Data. Each of these will be covered in later sections.

The editor is highly configurable, such that all windows can be detached from the main editor window, and left floating, pinned to a different area of the editor or even hidden from view.

5 Making a Basic Application

To demonstrate the use of the editor, it is best to show how to create a simple application from scratch.

The steps taken will be:-

1) Create a blank application
2) Specify the type of target to use
3) Create some containers and codeblocks
4) Wire the data pins together
5) Test the application

5.1 Create a blank application

Select File | New.

This will create a brand new blank application.

5.2 Specify the type of target to use

The Nano hardware is available in a range of different guises. For example, the analog board may not be fitted. Hence it is possible to specify the "type" of target intended for this application.

The basic types are:-

- CPU only
- Digital I/O only
- Analogue and Digital I/O

Other types may be available depending upon the installed version of Visual C||Cure.

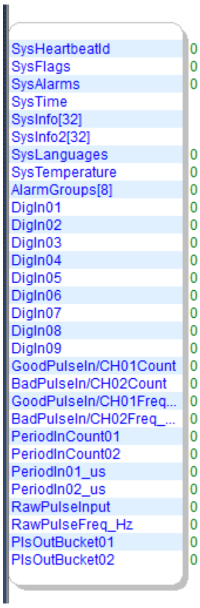
To specify the target type, select File | Properties.

In the "Target Device" list, select "Nano_Digital_Only", then click OK.

Note: the other properties shown (Password and Default Printer) will be covered later.

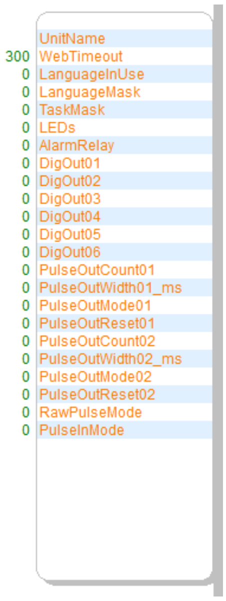
The main editor window will now show a list of pins on the left hand side, providing the top level container access to the hardware inputs from the target.

Scrolling the main editor window to the right will also show a list of pins on the right hand side. This shows the list of hardware outputs for the target.



SysHeartbeatId	0
SysFlags	0
SysAlarms	0
SysTime	
SysInfo[32]	
SysInfo2[32]	
SysLanguages	0
SysTemperature	0
AlarmGroups[8]	0
DigIn01	0
DigIn02	0
DigIn03	0
DigIn04	0
DigIn05	0
DigIn06	0
DigIn07	0
DigIn08	0
DigIn09	0
GoodPulseIn/CH01Count	0
BadPulseIn/CH02Count	0
GoodPulseIn/CH01Freq...	0
BadPulseIn/CH02Freq_...	0
PeriodInCount01	0
PeriodInCount02	0
PeriodIn01_us	0
PeriodIn02_us	0
RawPulseInput	0
RawPulseFreq_Hz	0
PlsOutBucket01	0
PlsOutBucket02	0

Hardware Inputs



	UnitName
300	WebTimeout
0	LanguageInUse
0	LanguageMask
0	TaskMask
0	LEDs
0	AlarmRelay
0	DigOut01
0	DigOut02
0	DigOut03
0	DigOut04
0	DigOut05
0	DigOut06
0	PulseOutCount01
0	PulseOutWidth01_ms
0	PulseOutMode01
0	PulseOutReset01
0	PulseOutCount02
0	PulseOutWidth02_ms
0	PulseOutMode02
0	PulseOutReset02
0	RawPulseMode
0	PulseInMode

Hardware Outputs

5.3 Create some containers and codeblocks

Now the application has some data to “use”, it can be manipulated in any way possible.

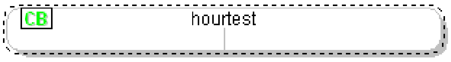
For this simple example, an application will be made to check the time and trigger a digital output pin on each hour.

Right click on the main editor window, and select New | New Codeblock.

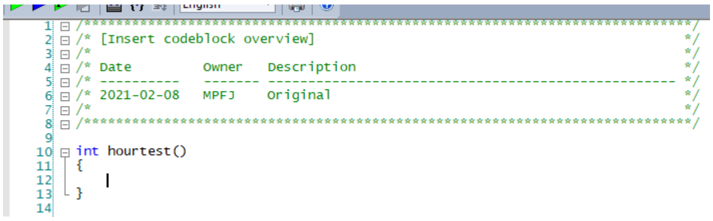
A new codeblock, called Codeblock will be made. The codeblock can be renamed at any time by changing this Name property in the properties window.

The codeblock can be moved around the editor window, as required, by left-clicking on the codeblock's name and dragging with the mouse.

In the properties window, change the name of the codeblock to “hourtest”.



Now double-click on the codeblock, and the editor window will change to show the C code editor, with a new comments section and empty C function called hourtest().



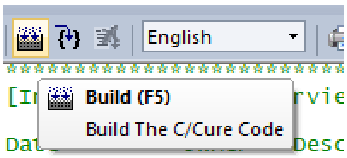
```
/****************************************************************************/
/* [Insert codeblock overview]                                              */
/*                                                                          */
/* Date         Owner    Description                                        */
/* ----------   -------  -------------------------------------------------- */
/* 2021-02-08   MPFJ     Original                                           */
/*                                                                          */
/****************************************************************************/

int hourtest()
{
    
}

```

The input and output parameters now need to be specified, so change the first line to:-

```
int hourtest(     double time,
                  int    &hour_trigger)
```

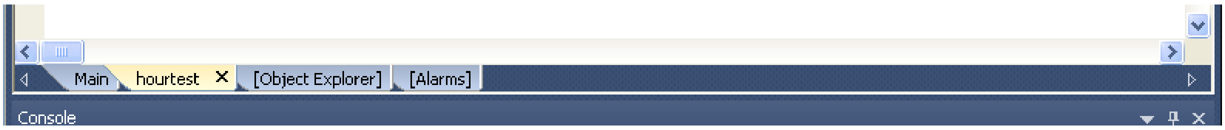
This specifies that our codeblock is to have one input (of type double) called time, and one output (of type int) called hour_trigger. The '&' symbol denotes the variable is to be an output variable.

Click on the Build icon (or press F5, or select Code View | Build), and the editor will check the syntax for any errors.

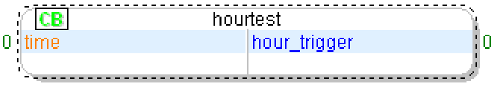


If any errors are found, they will be reported in red on the console and the cursor placed on the error. The check only reports the first error found, therefore the build check should be repeated to ensure no other errors exist.

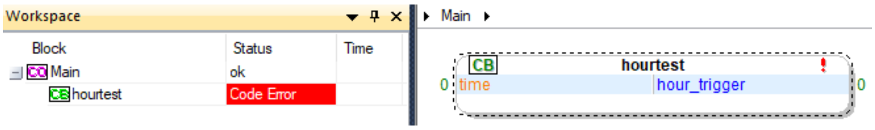
Once the codeblock builds without error, click back on the Main editor window using the tab strip at the bottom.



The codeblock will now have changed to reflect the new input / output parameters.



If any errors remain in a codeblock, the codeblock's status (as shown in the workspace tab) will be shown as “Code error”, and a red '!' indicator is shown, such as:-



Block	Status	Time
Main	ok	
hourtest	Code Error	

Now click back on the “hourtest” tab, enter the following code between the braces and click Build to check the syntax:-

```
{
      int minute = nftimeminute(time);
      if (minute == 0)
             hour_trigger = 1;
}
```

This codes uses the standard library routine nftimeminute() to return the minute value from an encoded date / time value (see 7.4 - Data types).

If this value is 0, then the output “hour_trigger” is set to a 1.

Note that there is no need to set the output pin to a 0, since, by default, all output pins are non-persistent, meaning they are automatically reset to zero at the start of each cycle (see 6.13 - Wrapped / Persistent / Unconnected Pins).

5.4 Wire the data pins together

Now that the application function has been written, the input and output pins need to be wired up.

Click on the Main tab, and then left-click and drag the “SysTime” hardware input and place it on the “time” input of the codeblock.

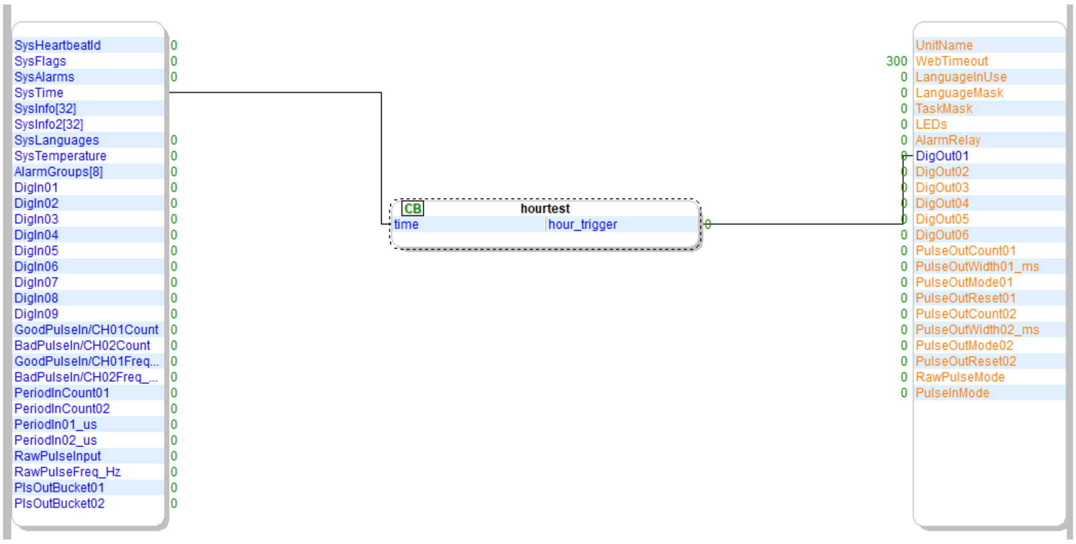
Note that when dragging a pin around, a “not allowed” symbol (⊘) will be shown whenever the pin cannot be connected. Pins can only be connected under the following conditions:-

a) only pins of the same data type and array size may be connected together (to ensure the predictability of the application)

b) input pins cannot be connected together, and output pins cannot be connected together

c) output pins can be connected to multiple input pins

Now click on the “hour_trigger” output pin and connect it to (say) the “DigOut01” hardware output.



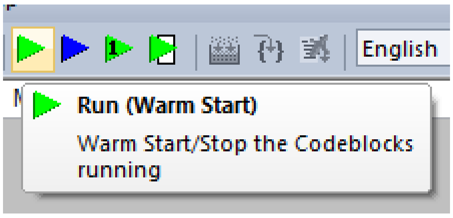
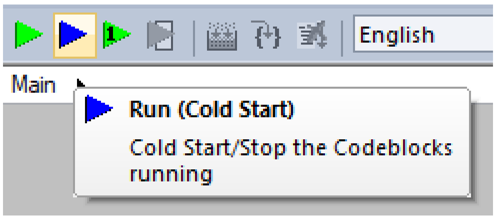
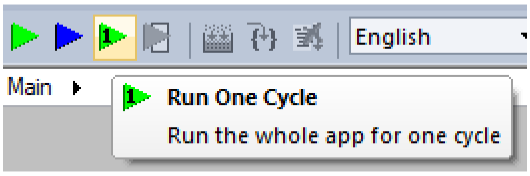
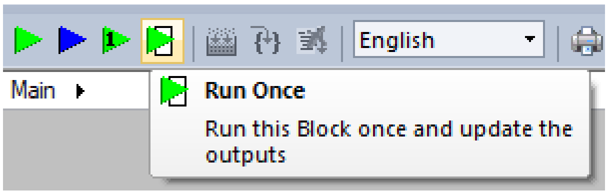
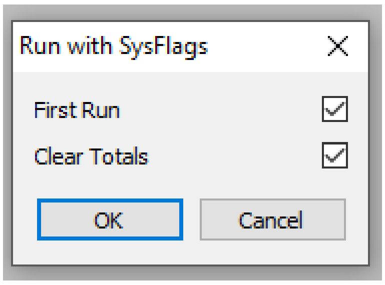
This now means that time input of the hourtest codeblock uses the data value output by “SysTime”, and “DigOut01” gets its value from the hour_trigger output.

5.5 *Test the application*

Another useful feature of the Visual C||Cure editor is the ability to run/debug an application without the need for physical hardware. There are some limitations to this as there are no “real” data to feed the hardware inputs, but the top level inputs can be manually set to simulate various test cases.

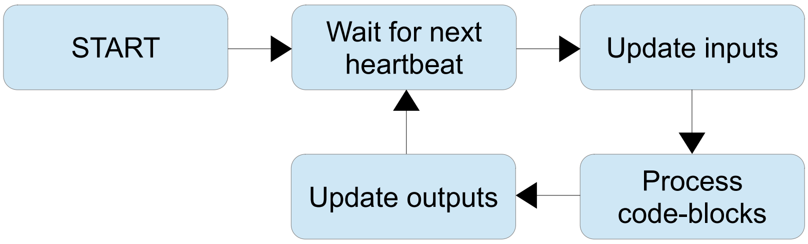
5.5.1 Run the entire application

The application can be run is one of four ways:-

Run (Warm Start) This starts running the application keeping all existing database values. Click the button again to stop the application.	 Run (Warm Start) Warm Start/Stop the Codeblocks running
Run (Cold Start) This starts running the application, but resets all database values to their "default" values. The "Cold Start" flag in the SysFlags pin is also set. Click the button again to stop the application.	 Run (Cold Start) Cold Start/Stop the Codeblocks running
Run One Cycle This runs the entire application for a single "warm" cycle.	 Run One Cycle Run the whole app for one cycle
Run Once This option is only available when a container or codeblock is selected. This runs the selected block for a single "warm" cycle.	 Run Once Run this Block once and update the outputs
Run with SysFlags... This option is only available from the menu Block View | Run with SysFlags... Choosing this option brings up a window allowing various bits of the SysFlags input to be set. Click OK to start the application.	 Run with SysFlags First Run ☑ Clear Totals ☑

When the application runs, the system performs the following actions for every calculation cycle:-

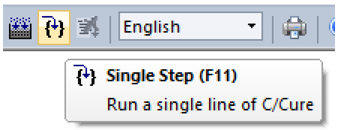
1) Wait for the next “heartbeat” (which occurs every 500ms).
2) Update the inputs to each codeblock / container.
3) Process the codeblocks.
4) Update the outputs from each codeblock / container.



So, for this example application, the top level inputs will update with the SysHeartbeatId input counting up from 0, and the SysTime input showing the current date and time. Then the hourtest codeblock is then processed, and, finally, the output pins are all updated.

Note that the data for any linked pins will also be updated on all containers and codeblocks, so the “time” input of the hourtest codeblock will also show the same as SysTime (since they have been connected together).

5.5.2 Codeblock Single Step



To debug a codeblock itself, it is possible to Single Step through the code.

To demonstrate, double-click on the hourtest codeblock, and then click on the Single Step button (or press F11, or select Code View | Step).

A blue rectangle to the left of the code indicates which line the debugger is currently on, and the right hand window changes to list all the variables of the current function.

Notice that the hour_trigger and minute variables are both at '0', as they are non-persistent.

Select Single Step again and the minute variable will be updated to reflect the current minute.

Single Step again and codeblock will either move to the end bracket (if minute is not zero), or on to the line which sets the hour_trigger output.

To simulate a new hour, the minute variable can be manually changed. Single step until the indicator is on the line “if (minute == 0)”, and then double-click on the minute variable in the right hand window. Enter a new value (i.e. zero) into the pop-up window, and click OK. This now overrides the previous value returned from nftimeminute(). Single stepping through will now run the “hour_trigger = 1” line, and the hour_trigger variable will be updated accordingly.

To stop debugging immediately, click on the Stop Debug button (or select Code View | Stop).

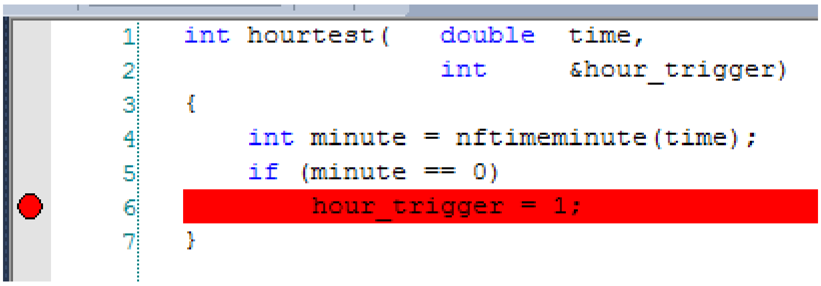
To run the codeblock to completion (or till the next breakpoint), press F8 (or select Code View | Run).

Note: it is also possible to single-step through codeblocks when not in Debug mode (the behaviour is identical apart from no live values are fed into the input pins). The input variables can still be manually changed, allowing codeblocks to be tested with any set of test inputs, and without the need for any hardware.

5.5.3 Codeblock Breakpoints

The debug engine also allows breakpoints to be set in codeblocks when the application is running locally.

To set a breakpoint, double-click on the grey margin of the code editor on the relevant line. A red circle will be shown and the line will be highlighted in red, as shown below:-



```
int hourtest(    double   time,
                 int      &hour_trigger)
{
    int minute = nftimeminute(time);
    if (minute == 0)
        hour_trigger = 1;
}
```

When the codeblock is running, the code will stop if a breakpoint is met. You can then single step (F11), continue (F8) or stop debugging as required.

Note: Breakpoints are not supported when in Remote Debug mode.

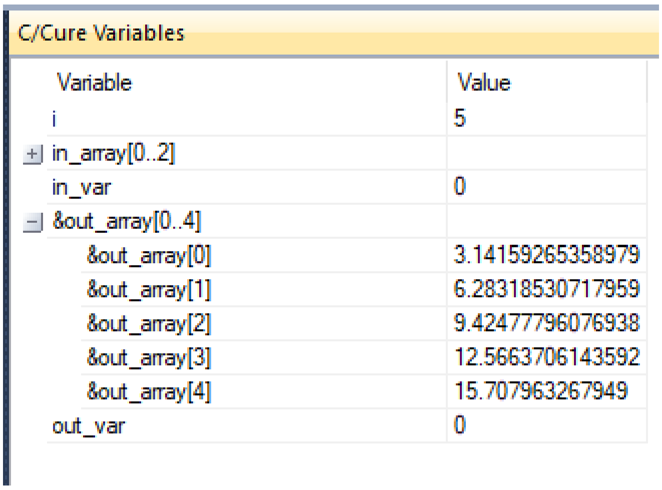
Note: Breakpoints are not stored in the application file when saved.

5.5.4 Codeblock Variables

When debugging a codeblock, the Properties pane shows the value of any variables.

Array variables are grouped and can be expanded/collapsed.

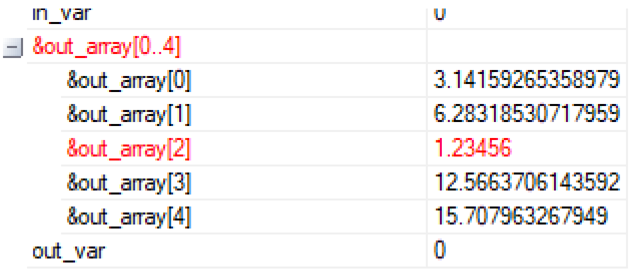
Output variables are prefixed with an ampersand symbol (&).



C/Cure Variables

Variable	Value
i	5
+ in_array[0..2]	
in_var	0
− &out_array[0..4]	
&out_array[0]	3.14159265358979
&out_array[1]	6.28318530717959
&out_array[2]	9.42477796076938
&out_array[3]	12.5663706143592
&out_array[4]	15.707963267949
out_var	0

On each single step, any variable value changes are highlighted in red.

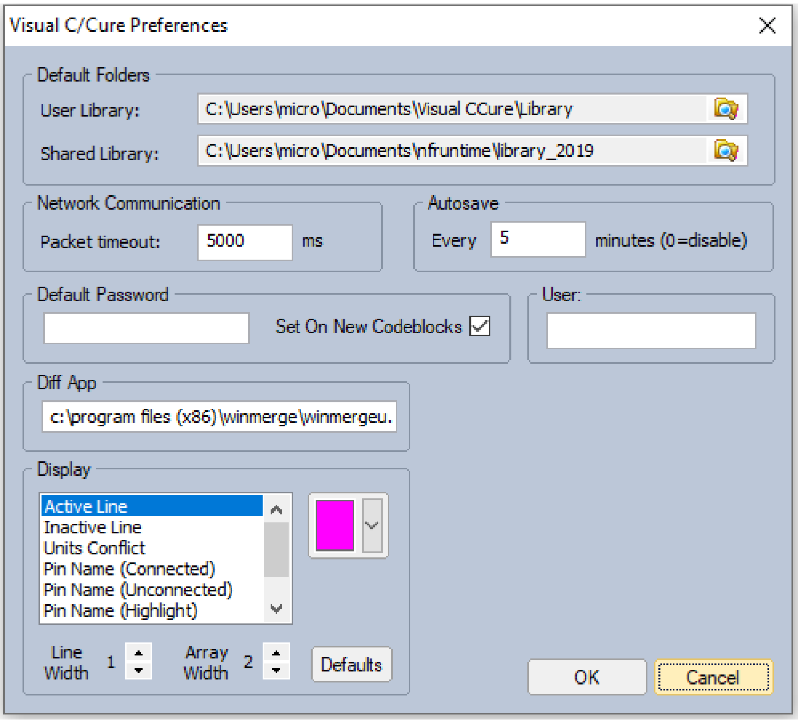


in_var	0
− &out_array[0..4]	
&out_array[0]	3.14159265358979
&out_array[1]	6.28318530717959
&out_array[2]	1.23456
&out_array[3]	12.5663706143592
&out_array[4]	15.707963267949
out_var	0

6 Editor Features

6.1 System Preferences

Various system wide preferences can be configured by selecting File | Visual CCure Preferences.



Visual C/Cure Preferences

Default Folders

User Library: C:\Users\micro\Documents\Visual CCure\Library

Shared Library: C:\Users\micro\Documents\nfruntime\library_2019

Network Communication

Packet timeout: 5000 ms

Autosave

Every 5 minutes (0=disable)

Default Password

Set On New Codeblocks ☑

User:

Diff App

c:\program files (x86)\winmerge\winmergeu.

Display

- Active Line
- Inactive Line
- Units Conflict
- Pin Name (Connected)
- Pin Name (Unconnected)
- Pin Name (Highlight)

Line Width 1

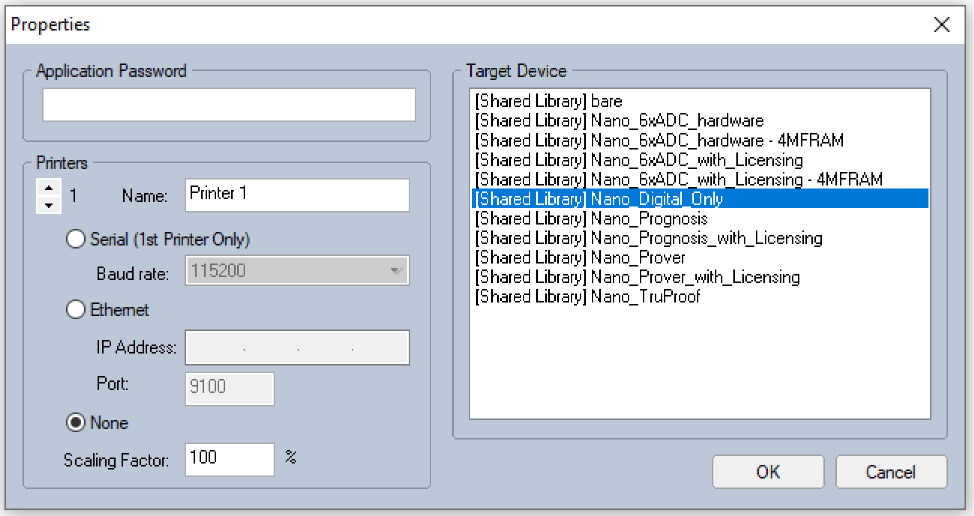
Array Width 2

Below is a list of the available preferences:-

Library Folder Paths	These define the User and Shared paths used by the library system. See 6.10 - Librarian for more information.
Network Communication	This sets the network timeout (in milliseconds) when attempting to communicate with remote targets.
Autosave	This sets the autosave interval (in minutes). A setting of 0 will disable the autosave feature.
Default Password	This sets the default security password. If the “Set On New Codeblocks” option is enabled, all newly created codeblocks with have this password assigned to them. See 6.11 - Security for more information.
User	Sets the username to be added to the optional log file entries when the application is saved.
Display	These settings allow various system colours and line widths to be customised.

6.2 Application Properties

Selecting File | Properties brings up the Properties dialog for the current application.



Properties

Application Password

Printers

1 Name: Printer 1

○ Serial (1st Printer Only)

Baud rate: 115200

○ Ethernet

IP Address: . . .

Port: 9100

● None

Scaling Factor: 100 %

Target Device

[Shared Library] bare
[Shared Library] Nano_6xADC_hardware
[Shared Library] Nano_6xADC_hardware - 4MFRAM
[Shared Library] Nano_6xADC_with_Licensing
[Shared Library] Nano_6xADC_with_Licensing - 4MFRAM
[Shared Library] Nano_Digital_Only
[Shared Library] Nano_Prognosis
[Shared Library] Nano_Prognosis_with_Licensing
[Shared Library] Nano_Prover
[Shared Library] Nano_Prover_with_Licensing
[Shared Library] Nano_TruProof

Here it is possible to specify the following:-

Application Password	This is the top-level application password. Once set, this password is required to load up the application into Visual C||Cure. This is not the same password as can be set for containers and codeblocks. See 6.7 - Codeblocks and 6.8 - Containers for more information.
Target Device	This list sets the top-level hardware inputs and outputs available to the application.
Printers	This allows the default printer settings to be specified, as follows:- • **Serial** : use an ASCII serial printer. • **Ethernet** : use a Postscript printer. • **None** : no default printer. The scaling factor is used for Ethernet printers. You can use the spin control to select between Printer 1, 2 and 3.

6.3 Panning / Scrolling

The main editor window can be scrolled left/right and up/down using the relevant scrollbar.

Alternatively, use the mouse scroll wheel to scroll the window up/down and Shift + scroll wheel to scroll left/right.

Finally, when dragging a container/codeblock or pin around the screen, when you reach the edge of the editor window, the view will automatically scroll up/down/left/right depending on which edge you are next to.

6.4 Zooming

The main editor window can be zoomed in and out to best fit the user's working preferences.

Simply hold down the Ctrl key and use the mouse wheel to zoom in and out. Alternatively, the window can be zoomed to “pre-defined” settings, as follows:-

Ctrl+1	Actual Size	resets to 100% zoom
Ctrl+2	Fit Page	sets the zoom level to show all containers and codeblocks
Ctrl+3	Fit Width	sets the zoom such the entire width of the application can be viewed

All these functions are also available through the View | Zoom menu.

6.5 Resizing

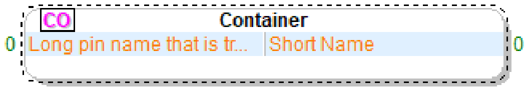
The “sheet size” of the application can be adjusted.

Use the scroll bars to show the right or bottom edge of the white page in the main window. Hover the mouse over the white / grey line and the cursor will change to a left/right or up/down arrow. Now click and drag the page to increase / decrease the page size.

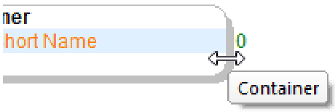
The bottom right corner also acts as a resize point, allowing the page to be resized in both directions at once.

Note that each container has its own adjustable sheet size.

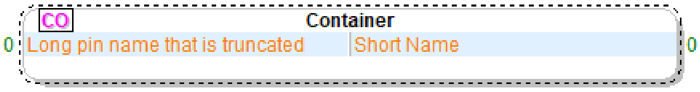
The width of any container or codeblock can also be adjusted. Consider the following container:-



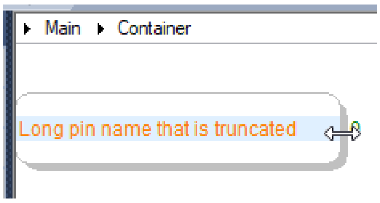
If you hover of the left or right edge, the resize cursor will appear.



You can then resize the container as required:-



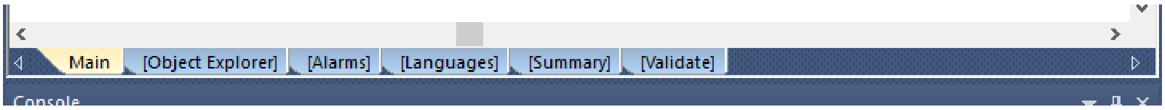
For containers, you can also resize the Input and Output blocks within the container to achieve the same result.



Again, each container and codeblock can be resized independently.

6.6 Editor Tabs

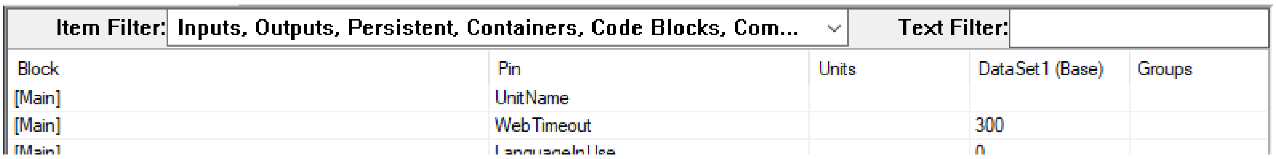
At the bottom of the main editor window is a tab-strip, showing the currently opened codeblocks and containers as well as various “system” tabs as explained below.



These tabs can be used to view the relevant codeblock or container, and the item currently being viewed can be closed by clicking on the 'X' icon next to the item's name.

6.6.1 [Object Explorer]

The Object Explorer tab gives an overview of all input pins, output pins and comments within the application. The list of items can be filtered by item type and text filter.



Item Filter: Inputs, Outputs, Persistent, Containers, Code Blocks, Com...

Text Filter:

Block	Pin	Units	DataSet1 (Base)	Groups
[Main]	UnitName			
[Main]	WebTimeout		300	
[Main]	LanguageInUse		0	

The following columns are shown:-

Block	Shows the hierarchical pathname to the block. Double-clicking on the name will load up the relevant container and highlight the relevant block.
Pin	Shows the pin name. Arrays items are individually listed. Double-clicking on the pin name will load up the relevant container and highlight the relevant pin.
Units	Shows the current units value for the pin. Double-clicking on the cell will bring up a list of units (as defined in the UNITS table) and the user can pick a new unit value if required. If the pin's unit value is linked to the units table, rather than just being hard-coded text, an asterisk (*) will be shown at the end of the unit value.
Dataset(s)	Shows the current default value for a persistent pin. Double-clicking on the item will allow the user to change the value. This item will be greyed out for all output pins. Extra columns can be added/removed/renamed/cloned by right-clicking on the header strip. By default, all pins as set to copy the value from the base dataset (denoted by <Base>). However, each dataset can have its own set of default values for each input pin, and double-clicking on the item will allow the user to change the value as required. In the value edit window, clicking on the “Set to Base Value” button will reset the value back to <Base>.
Groups	Shows the current groups the pin is a member of. See 6.12 - Pin Properties for more information on groups.

The Item Filter can be used to limit what type of pins are shown. The following options are available:-

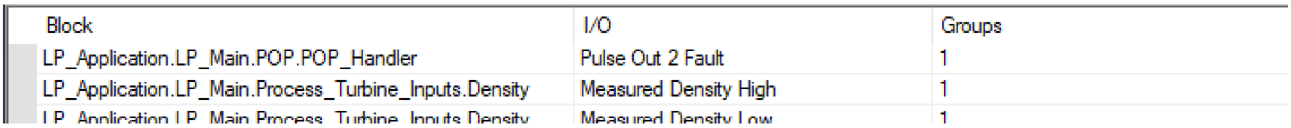
Inputs	Show input pins.
Outputs	Show output pins.
Non Persistent	Show non-persistent pins.
Persistent	Show persistent pins.
Wrapped	Show wrapped pins.
Containers	Show container pins.
Code Blocks	Show code block pins.
Comms Blocks	Show comms link pins.
Comments	Show comments.

The Text Filter can be used to show only pins or comments that include specific text.

Note that the Item Filter and Text Filter work in conjunction with each other so, for example, selecting only “Comments” in the Item Filter and entering “abc” in the Text Filter will only show Comments containing the text “abc”, and no other items.

6.6.2 [Alarms]

The Alarms tab gives an overview of all the output pins that have been configured as an alarm source.

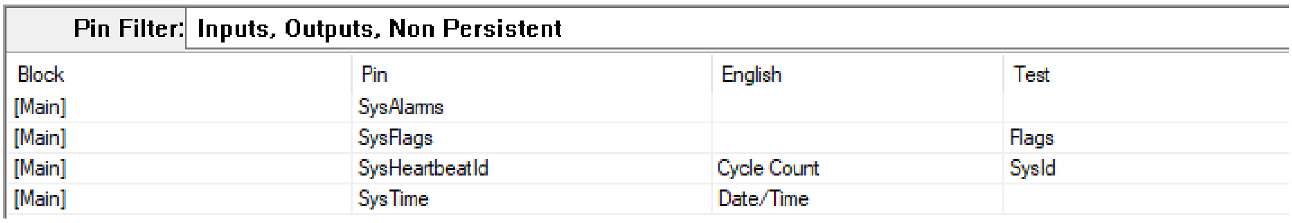


Block	I/O	Groups
LP_Application.LP_Main.POP.POP_Handler	Pulse Out 2 Fault	1
LP_Application.LP_Main.Process_Turbine_Inputs.Density	Measured Density High	1
LP_Application.LP_Main.Process_Turbine_Inputs.Density	Measured Density Low	1

Whenever any such pin is non-zero, an alarm will be raised and logged in the Alarm Log. See 8.6 - Alarms for more information.

6.6.3 [Languages]

The Languages tab shows all pins in the application, a column for each configured language plus the current text alias for each pin. For example:-



Pin Filter: Inputs, Outputs, Non Persistent

Block	Pin	English	Test
[Main]	SysAlarms		
[Main]	SysFlags		Flags
[Main]	SysHeartbeatId	Cycle Count	SysId
[Main]	SysTime	Date/Time	

Languages can be added/removed/renamed by right-clicking on the header strip.

By default, a new language has blank entries for all pins.

Double-clicking on a pin/language cell allows the user to change the pin text alias for that language.

Leaving an entry blank allows the text alias to "default" to either the base language or the pin name itself (if no text alias exists in the base language).

From the example above, the following text aliases would be used:-

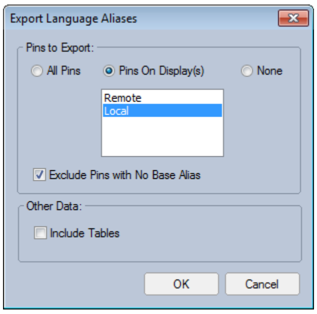
Pin	English	Test
SysAlarms	SysAlarms	SysAlarms
SysFlags	SysFlags	Flags
SysHeartbeatId	Cycle Count	SysId
SysTime	Date/Time	Date/Time

Similar to the Object Explorer, the pins can be filtered by type or string search.

Importing and exporting of language data is also possible (via the right-click menu).

Data is exported as a CSV file and the user can chose to export all pins, or only those contained in a particular display tree. Pins with no base alias can also be excluded.

The CSV file can then be edited and imported back into the application to allow bulk updating of the pins aliases.



Export Language Aliases

Pins to Export:

○ All Pins ● Pins On Display(s) ○ None

Remote
Local

☑ Exclude Pins with No Base Alias

Other Data:

□ Include Tables

6.6.4 [Summary]

The Summary tab shows the current release notes for the application. These notes can be edited and any changes will be stored next time the application is saved.

6.6.5 [Validate]

The Validate tab provides an overview of an issues detected in the design of the application. This might include, for example, invalid or missing pin references, library block issues or report resource usage.

The description column gives an explanation of the check. **RED** is used to denote an error, **ORANGE** for a warning, **GREEN** means no issue has been found and **BLACK** means information only.

In most instances, double-clicking on an item will navigate to the relevant item in the application.

Below lists the sections checked and the possible messages:-

6.6.5.1 Alarms

Duplicate alarm name	The alarms shown in the Info boxes have the same name, which may be confusing.

6.6.5.2 Comms

Invalid comms name	The name of the comms link contains invalid characters which would prevent it from being saved to the library.
No datamap has been selected	The comms link has been designed to have a datamap assigned to it but none has been selected.
Invalid start address	The poll shown in the Info #1 column has an invalid start address.
Invalid address count	The poll shown in the Info #1 column has an invalid address count.
Invalid address(es) in use	The poll shown in the Info #1 column references an address that does not exist in the comms link datamap.
Invalid pin name Invalid pin reference for <x> Missing dynamic pin group <x> Units conflict	See relevant entry in 6.6.5.3 - Containers / Codeblocks.

6.6.5.3 Containers / Codeblocks

Invalid pin name	The pin shown contains invalid characters.
Invalid pin reference for ADP	The pin shown in the Info #2 column references an ADP lookup pin that does not exist. The Info #1 column gives the path of the missing pin.
Invalid pin reference for UNITS	The pin shown in the Info #2 column references a UNITS lookup pin that does not exist. The Info #1 column gives the path of the missing pin.
Library: File missing	The block shown in the Info #2 column references a library file that does not exist. The Info #1 column shows if the 'record checksum' property is set and the current checksum of the block.
Library: In sync	The block shown in the Info #2 column is in sync with the library file. The Info #1 column shows if the 'record checksum' property is set and the current checksum of the block.

Library: Out of sync	The block shown in the Info #2 column is out-of-sync with the library file. The Info #1 column shows if the ‘record checksum’ property is set and the current checksum of the block.
Missing dynamic pin group ADP	The pin shown in the Info #2 column references an ADP lookup pin. However, the ADP pin itself does not have the ADPSELECT group ticked. The Info #1 column gives that to the invalid ADP pin.
Missing dynamic pin group UNITS	The pin shown in the Info #2 column references a UNITS lookup pin. However, the UNITS pin itself does not have the UNITSELECT group ticked. The Info #1 column gives that to the invalid UNITS pin.
Units conflict	The pin shown in the Info #2 column has a units setting that conflicts with another pin.

6.6.5.4 Datamaps

Invalid pin reference	The address shown in the Info #2 column uses a pin that does not exist. The Info #1 column gives the path of the missing pin.
Invalid array index	The address shown in the Info #2 column uses an invalid array index. The Info #1 column gives the path of the missing pin.
Duplicate pin	The pin (and possible array index) shown are used multiple times in the datamap. This may be intentional so this is only flagged as a warning. Note that any invalid pins are not tested for duplicates.
Duplicate tagname	The tagname show is duplicated. The address of first use is shown in the Info #2 column. Note that this condition is only checked for in the specific TAGNAMES datamap.

6.6.5.5 Displays

Invalid pin reference	The display item shown in the Info #2 column uses a pin that does not exist. The Info #1 column gives the path of the missing pin.
Invalid array index	The display item shown in the Info #2 column uses an invalid array index. The Info #1 column gives the path of the missing pin.
Invalid Ident pin reference	The display item shown in the Info #2 column uses a non-existent IDENT pin. The Info #1 column gives the path of the missing pin.
Invalid Ident array index	The display item shown in the Info #2 column uses an invalid array index for the IDENT pin. The Info #1 column gives the path of the missing pin.
Invalid Active pin reference	The display item shown in the Info #2 column uses a non-existent ACTIVE pin. The Info #1 column gives the path of the missing pin.

Invalid Active array index	The display item shown in the Info #2 column uses an invalid array index for the ACTIVE pin. The Info #1 column gives the path of the missing pin.
Invalid Active value	The display item shown in the Info #2 column uses a table on the ACTIVE pin but the active value does not exist in the specified table. The Info #1 column gives the path of the missing pin.

6.6.5.6 Historicals

Invalid pin reference	The index shown in the Info #2 column uses a pin that does not exist. The Info #1 column gives the path of the missing pin.
Invalid array index	The index shown in the Info #2 column uses an invalid array index. The Info #1 column gives the path of the missing pin.
Invalid Trigger pin reference	The TRIGGER pin does not exist. The Info #1 column gives the path of the missing pin.

6.6.5.7 Reports

The first item in the Reports section is “Zone Info”. It simply shows all the report zones along with a list of all reports that use that zone.

Missing pin reference	The placement shown in the Info #2 column does not have a pin defined.
Invalid pin reference	The placement shown in the Info #2 column uses a pin that does not exist. The Info #1 column gives the path of the missing pin.
Invalid Active pin reference	The ACTIVE pin of the placement shown in the Info #2 column uses a pin that does not exist. The Info #1 column gives the path of the missing pin.
Invalid Line Active pin reference	The ACTIVE pin of the report line shown in the Info #2 column uses a pin that does not exist. The Info #1 column gives the path of the missing pin.
Invalid Trigger pin reference	The TRIGGER pin does not exist. The Info #1 column gives the path of the missing pin.
Invalid Visible pin reference	The VISIBLE pin does not exist. The Info #1 column gives the path of the missing pin.
Report uses too many data slots	A report can store a maximum of 240 data “slots”, where each slot can store 1 double, 1 integer or 8 bytes of a string. The data configured for the report has been calculated to exceed this limit and so not all data will be stored. The Info #1 column gives details of the slot usage.
Trigger pin not set	The TRIGGER pin for the report has not been set.

6.6.5.8 Resource Usage

This section details the database resource usage, showing how many DOUBLE and INTEGER variables are currently used by the application.

Non-volatile variables are listed separately (as these are stored in FRAM which has limited capacity), along with a percentage usage in relation to the hardware target chosen for the application.

If the non-volatile usage is above 90% this is classed as a “warning” and highlighted in **ORANGE**.

If the non-volatile usage is above 100% this is classed as an “error” and highlighted in **RED**.

The number of alarms defined is also shown.

6.6.5.9 Tables

Unused table	The table is not referenced by any pin in the application.

6.7 Codeblocks

Codeblocks allow developers to add C code to their applications.

Creating a new codeblock can be done by right-clicking on the main editor window and selecting New | New Codeblock (or pressing 'c' or selecting Edit | New Block | New Codeblock).

Codeblocks can be deleted, copied and pasted, and stored in the library.

The ordering of the input and output pins can be adjusted (to help beautify the application wiring) by right-clicking on the relevant pin and selecting Move Up (or pressing 'u' or File | Edit | Move Up) or Move Down (or pressing 'd' or File | Edit | Move Down).

See 7 - C||Cure Syntax for more information on the format and coding style used by codeblocks.

A list of currently opened codeblocks (and containers) is shown in the tab-strip at the bottom of the main editor window.

The following codeblock properties are available on the Properties window:-

Name	Name of the codeblock.
Library	Shows the library this codeblock was loaded from (if any).
Library Path	Shows the library path and filename for this codeblock.
Version (optional)	If this codeblock has been stored in the library, this shows the version of the codeblock.
Checksum (optional)	If this codeblock has been stored in the library, this shows the current md5 checksum.
Record Checksum (optional)	If checked, this codeblock and its checksum will be listed in the Constants Log.
Password	Allows the codeblock's password to be changed
Timeout (ms)	Shows the current timeout before the codeblock will be forcefully stopped from executing. This can be used to prevent any single set of code from consuming too much processing time.

6.8 Containers

Containers allow a hierarchical structure to be used within an application.

They can contain either codeblocks, other containers, or a mixture of the two.

Creating a new container can be done by right-clicking on the main editor window and selecting New | New Container (or pressing 'C' or selecting Edit | New Block | New Container).

This will create a new empty container, with no input or output pins.

The default container name will be “Container”, or, if such a container already exists, “Container_X” where X is a number chosen not to clash with any other containers at the current hierarchical level.

Input and output pins can be added to the current container by right-clicking on any empty space in the editor window, and selecting New | New Input Pin or New Output Pin (or pressing 'i' or 'o' or selecting Edit | New Input Pin or New Output Pin).

Similarly, input and output pins can be added to a specific container by right-clicking on the required container, and selecting New | New Input Pin or New Output Pin.

Similar to codeblocks, the ordering of the input and output pins can be adjusted by right-clicking on the relevant pin and selecting Move Up (or pressing 'u' or File | Edit | Move Up) or Move Down (or pressing 'd' or File | Edit | Move Down).

Double-clicking on a container will drill down into that section of the application, with a list of currently opened containers (and codeblocks) shown at the bottom of the main editor window.

Container pins can be edited as described in 6.12 - Pin Properties.

The following container properties are available on the Properties window:-

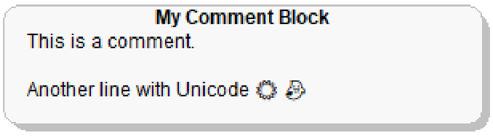
Name	Name of the container.
Library	Shows the library this container was loaded from (if any).
Library Path	Shows the library path and filename for this container.
Version (optional)	If this container has been stored in the library, this shows the version of the container.
Checksum (optional)	If this container has been stored in the library, this shows the current md5 checksum.
Record Checksum (optional)	If checked, this container and its checksum will be listed in the Constants Log.
Password	Allows the container's password to be changed.
Iterations	Specifies how many times the container should be run per application cycle.
Iteration Control	Specifies an output which, when non-zero, will abort the above iterations.

6.9 Comments

Free text comment blocks can be added to any container.

Creating a new comment block can be done by right-clicking on the main editor window and selecting New | New Comment (or pressing 'm' or selecting Edit | New Block | New Comment).

The comment name and comment text can be changed using the Properties window.



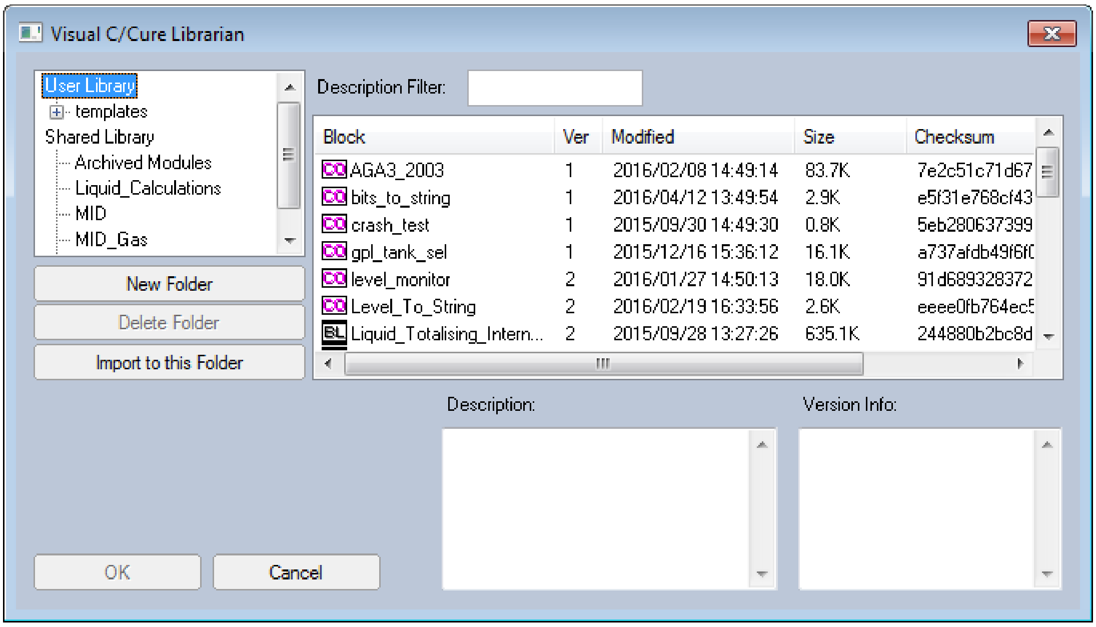
My Comment Block

This is a comment.

Another line with Unicode ⚙

6.10 Librarian

Visual C||Cure has the ability to store codeblocks and containers in a library, so that they may be re-used within the same application, or in completely different applications altogether.



Description Filter:

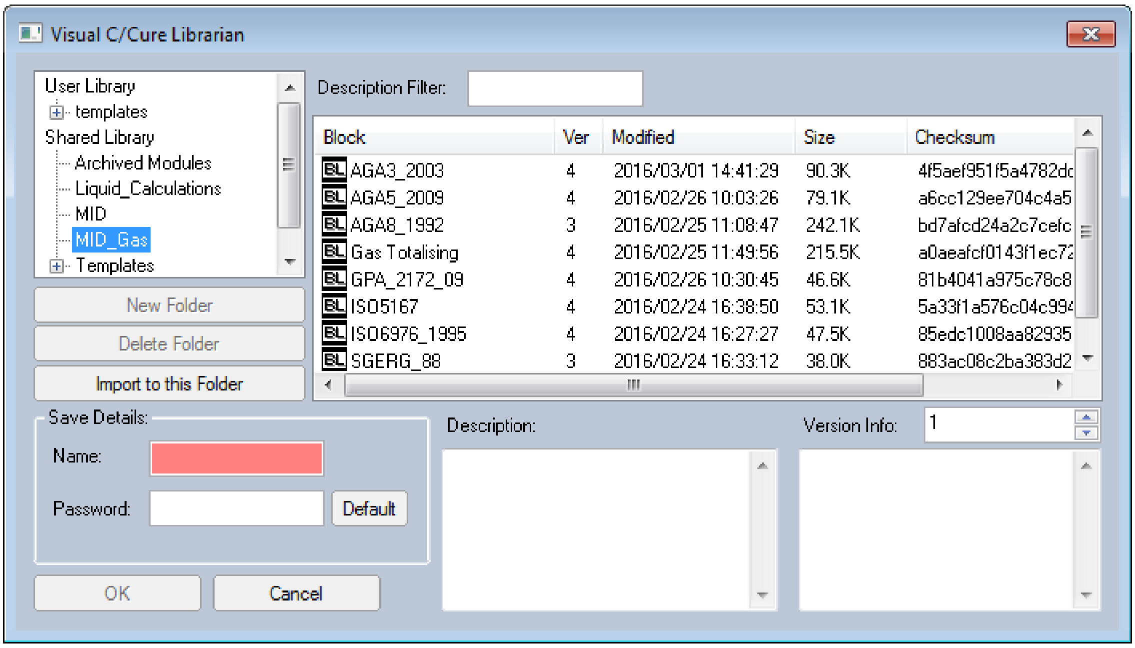
Block	Ver	Modified	Size	Checksum
CO AGA3_2003	1	2016/02/08 14:49:14	83.7K	7e2c51c71d67
CO bits_to_string	1	2016/04/12 13:49:54	2.9K	e5f31e768cf43
CO crash_test	1	2015/09/30 14:49:30	0.8K	5eb280637399
CO gpl_tank_sel	1	2015/12/16 15:36:12	16.1K	a737afdb49f6f(
CO level_monitor	2	2016/01/27 14:50:13	18.0K	91d689328372
CO Level_To_String	2	2016/02/19 16:33:56	2.6K	eeee0fb764ec5
BL Liquid_Totalising_Intern...	2	2015/09/28 13:27:26	635.1K	244880b2bc8d

The left hand side shows a tree view of the library folders. Folders can be added / removed from the User Library (see 6.1 - System Preferences).

Library blocks can be imported into any of the library folders using the "Import to this Folder" button.

6.10.1 Saving to the library

Once a codeblock or container has been written, it can be saved to the library by right-clicking on the codeblock / container and selecting Save | Link To Library.



Description Filter:

Block	Ver	Modified	Size	Checksum
BL AGA3_2003	4	2016/03/01 14:41:29	90.3K	4f5aef951f5a4782dc
BL AGA5_2009	4	2016/02/26 10:03:26	79.1K	a6cc129ee704c4a5
BL AGA8_1992	3	2016/02/25 11:08:47	242.1K	bd7afcd24a2c7cefc
BL Gas Totalising	4	2016/02/25 11:49:56	215.5K	a0aeafcf0143f1ec72
BL GPA_2172_09	4	2016/02/26 10:30:45	46.6K	81b4041a975c78c8
BL ISO5167	4	2016/02/24 16:38:50	53.1K	5a33f1a576c04c994
BL ISO6976_1995	4	2016/02/24 16:27:27	47.5K	85edc1008aa82935
BL SGERG_88	3	2016/02/24 16:33:12	38.0K	883ac08c2ba383d2

Version Info: 1

Select the required library directory on the left side of the window. Folders can be added or removed using the "New Folder" and "Delete Folder" buttons, and library blocks can be imported into any of the library folders using the "Import to this Folder" button.

Enter the name under which to save the item.

Alternatively, an existing library item can be selected if the user wishes to over-write the existing item with the codeblock / container to be saved.

At this point, a password can be applied to the item to be saved. The “Use Default” button will copy the password as set in the system preferences dialog (see 6.1 - System Preferences for more information).

The item version can also be adjusted. By default, the version will be 1 for new library items, or one more than the current version for existing library items.

An optional description and version summary can also be added at this point.

Click OK to save the item to the library.

If no such item already exists in the library, the application item will now be saved to the library.

However, if the item name matches an existing library item, the user will be prompted if they wish to sync with the existing unit. Selecting Yes will replace the application item with a linked copy of the existing library item, rather than over-writing the library item.

Selecting No will then prompt the user to confirm they wish to over-write the library item.

6.10.2 Loading from the library

Once stored in a library, items can be added to the current container by right-clicking on the editor window and selecting Load Block from Library (or pressing 'l' or selecting Edit | New Block | Load Block from Library...). This will bring up the librarian window.

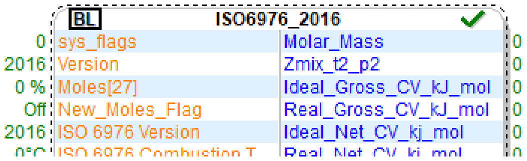
Select the required library directory on the left side of the window, and select the required library item from the right-hand list.

Click OK to add a new copy of the library item to the application.

6.10.3 Sync Status

When added to an application, a copy of the library item is used, but a link to the library is maintained and is used to check that the application item and the library item are still in sync.

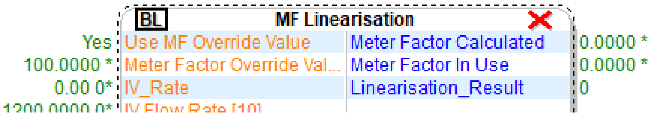
A green "✓" symbol, denotes that the application item and library item are in sync.



	BL ISO6976_2016 ✓		
0	sys_flags	Molar_Mass	0
2016	Version	Zmix_t2_p2	0
0 %	Moles[27]	Ideal_Gross_CV_kJ_mol	0
Off	New_Moles_Flag	Real_Gross_CV_kJ_mol	0
2016	ISO 6976 Version	Ideal_Net_CV_kj_mol	0
0°C	ISO 6976 Combustion T	Real_Net_CV_kj_mol	0

If the library item is ever updated, or the user changes the copy loaded into the application, the relevant codeblock or container will be flagged as "out of sync".

This "out of sync" information is shown in the Workspace tab (under the Status column) as well as on the codeblock / container itself (by a red "**X**" in the top right corner of the item):-

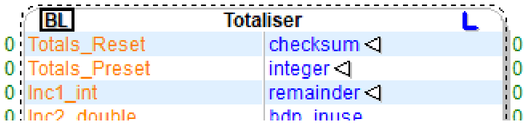


	BL MF Linearisation ✗		
Yes	Use MF Override Value	Meter Factor Calculated	0.0000 *
100.0000 *	Meter Factor Override Val...	Meter Factor In Use	0.0000 *
0.00 0*	IV_Rate	Linearisation_Result	0
1200.0000 0*	IV Flow Rate [10]		

The user then has the option to re-sync the application codeblock / container with the version located in the library. This can be achieved by right-clicking on the container or codeblock and selecting Update From Library.

This allows bug fixes and updates to commonly used code to be easily detected and updated.

One further status indicator is a blue "**L**" symbol.



	BL Totaliser L		
0	Totals_Reset	checksum ◁	0
0	Totals_Preset	integer ◁	0
0	Inc1_int	remainder ◁	0
0	Inc2_double	hdn_inuse	0

This indicates the block is linked to a library file but the file is missing from the library.

6.11 Security

To protect developers' Intellectual Property, several levels of security are available for each application:-

File encryption	All application files are encrypted when stored on either the PC or within the target. This encryption is based on the widely adopted AES standard (http://en.wikipedia.org/wiki/Advanced_Encryption_Standard).
Application password	By default, new applications are not protected, allowing anyone to open the application using Visual C||Cure. Using the application properties dialog, the developer can set a password that must be entered to allow Visual C||Cure to access the application when loaded. See 6.2 - Application Properties for more information.
Codeblock password	Each codeblock can be configured to have its own password. A default password can be set, and all new codeblocks can be configured to use this password by default, using the Visual C||Cure Preferences dialog (see 6.1 - System Preferences). If a codeblock has a password applied to it, and that password matches the default password, then it will be automatically viewable. However, if the passwords do not match, then the letters "BL" (meaning BLOCKED) will be shown in the top left corner of the codeblock. Double-clicking on the codeblock will prompt for the password to be entered. The password can be added or changed using the Password property in the Properties window.
Container password	Like codeblocks, each container can also be configured to have its own password. Passwords can be added or changed in the same way as codeblocks.

6.12 Pin Properties

All input and output data pins can be configured to suit the application being written.

Below is a definition of all the available properties, as shown in the Properties window:-

Name	Name of the pin (as defined by the input/output pins of the codeblock or container).
Languages	Text alias strings for each configured language. The user can edit the aliases here or via the [Languages] system tab (see 6.6.3 - [Languages]).
Data Type	Sets the data type for the pin:- **double** floating point number **int** whole number **text** fixed length character string See 7.4 - Data types for further information.
Base Type [read-only]	Shows the basic data type of this pin. This is the value to specify when feeding this pin into a codeblock.

Text Length	Shows the maximum string size of a text item, or of each text element if this pin is an array. This property is greyed out if the data type is not set as Text.
Dimension	A value of 1 denotes a single data point. Any value > 1 will convert the data point into a data array, with the dimension value setting the number of items in the array.
Units	Sets the units associated with the pin. Choosing <inherit> will set the pin to inherit the units from any connected pins. All entries in the UNITS table will also be listed. Any pins in the VCC_UNITSELECT group (see Groups below) will also be listed. Selecting any one of these will set the pin to use the value of the units pin. See 6.15 - Units for further information.
Default ADP	Sets the default number of digits shown “After the Decimal Place”. This does not affect the calculations, only the displaying of data values. **<inherit>** set the pin to inherit the ADP from any connected pins **0...15** show 0...15 decimal places **<variable>** set the pin to show up to 15 decimal places, but strip off any trailing zeros Any pins in the VCC_ADPSELECT group (see Groups below) will also be listed. Selecting any one of these will set the pin to use the value of the ADP pin. See 6.16 - ADP for further information.
Alarm	Specifies if this pin is to act as an alarm source. **Inactive** pin is not an alarm source **Active** an alarm will be set when the pin is non-zero Any pins in the VCC_ALARMSELECT group (see Groups below) will also be listed. Selecting any one of these will set the pin to be Active, but only trigger an alarm when the selected alarm pin is non-zero. See 8.6 - Alarms for more information.
Alarm Groups	Eight alarm groups are available, and are used to combined alarms. When any alarm is set, its corresponding Alarm Group bit is also set to 1. These bits are then presented as a top-level input array and can be used in the application as required. For example, there may be 10 alarms defined in the application. A digital output needs to be set when any one of 5 particular alarms are set. In which case, select Alarm Group 1 on the required alarms, and then wire the AlarmGroup[0] input to DigOut1. If any of the 5 alarms get set, then the Alarm Group 1 bit will be set, resulting in Digital Output 1 being set.
Width	Sets the total number of visible characters shown when displaying the data pin value. Text will be right justified, with spaces used to fill any leading characters (see also “Leading Zeros” option below). A value of 0 will show as many characters as necessary to view the full data value.
Leading Zeros	When used with the Width option above, this will use zeros (rather than spaces) to pad any leading characters when displaying data values.

Wrap	Specifies that an output pin will be fed back as an input pin for the next cycle. See 6.13 - Wrapped / Persistent / Unconnected Pins for more information. This property is greyed out for codeblock pins and container input pins.
Reset to X	This option only applies to persistent pins, otherwise it is greyed out. Selecting this option will cause the pin to be reset to a fixed value (see below) at the end of each cycle. This can useful when a one-off trigger needs to be activated (e.g. to run a user initiated report).
Reset Value	When "Reset to X" is checked, this sets the reset value.
Table	Specifies the string lookup Table to be used for this pin, causing the pin value to be shown as a string rather than a number. See 6.14 - Tables for more information.
Field Table	If the pin is an array (Dimension > 1) this specifies the string lookup Table to be used for this pin's fields. See 6.14 - Tables for more information.
Groups	Lists the available system / user-defined groups that can be selected for this pin. Groups may be added via the “GROUPS” table. See 6.14 - Tables for more information.
Value	Shows the current / default value(s) for this pin. If the pin is an array, then each array item will be listed. When running in DEBUG mode, the live pin values are displayed here.
Date/Time	This option only applies to pins with a “double” data type. Specifies that this data pin holds an encoded date/time value and should be displayed as such.
IP Address	Specifies that this data pin holds an IP address and should be displayed as such. Note that the data type must be a 4 element integer array.
Persistent	Specifies (or shows) whether this pin's data is persistent. See 6.13 - Wrapped / Persistent / Unconnected Pins for more information.

6.13 Wrapped / Persistent / Unconnected Pins

To guarantee predicable operation, the C||Cure computing engine is modelled on the behaviour of state machines, in that the output values are solely determined by the input values. All previous values of any outputs are forgotten, and so any fixed set of input values will produce the same set of output values, regardless of any previous calculations.

To achieve this all output pins on any container or codeblock are, by default, reset to zero at the start of each computation cycle.

Outputs will only be changed if the application code specifically changes them via a codeblock.

However, some values (e.g. user specified inputs, counters, etc) do need to be “remembered” between cycles. To cater for this, two types of pin option can be set, as follows:-

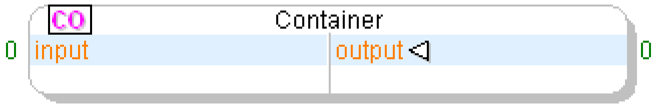
6.13.1 Wrapped

Wrapped pins only apply to container outputs, and are used when container output data values need to be copied from one cycle to the next.

Selecting the “Wrap” option on an output pin of a container will generate a dummy input pin.

At the end of each computation cycle, the values of any wrapped output pins are then copied to the relevant dummy input pin, thus allowing the output data to be used any as input data on the next cycle.

Wrapped pins are denoted by a “◁” symbol, as shown on a container symbol:-



… or within the container itself:-

input
output ◁
0
0
0
output ◁

6.13.2 Persistent

To complement wrapped pins, persistent pins do not get reset to zero at the start of each cycle.

The “Persistent” option is always set (and cannot be changed) for unconnected input pins (i.e. those that are not fed by another pin and can only be changed through a comms link or the web interface).

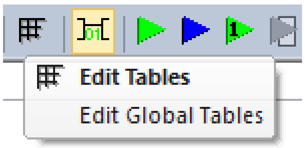
Persistent data is stored in non-volatile memory and, therefore, the data values are retained when the unit is power cycled.

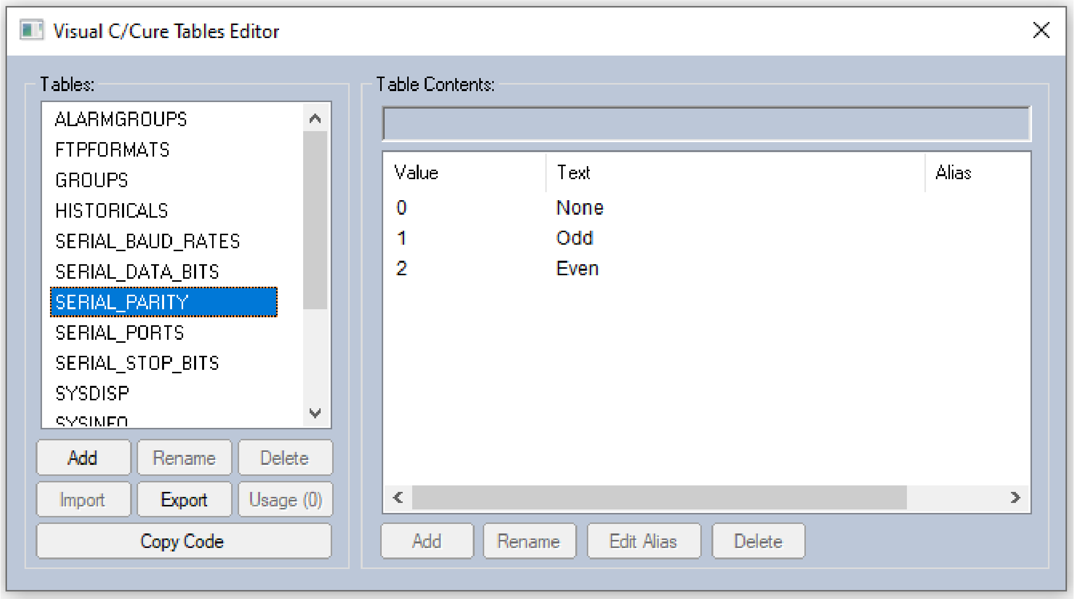
Note that non-volatile memory is limited so this should be taken into consideration when designing applications. In many cases, a wrapped pin can be used in place of a persistent pin.

6.14 Tables

To allow more human readable content to be applied to an application, string lookup tables can be configured.

Click on the “Edit Tables” icon (or select Edit | Edit Tables), which brings up the Tables Editor:-





Value	Text	Alias
0	None	
1	Odd	
2	Even	

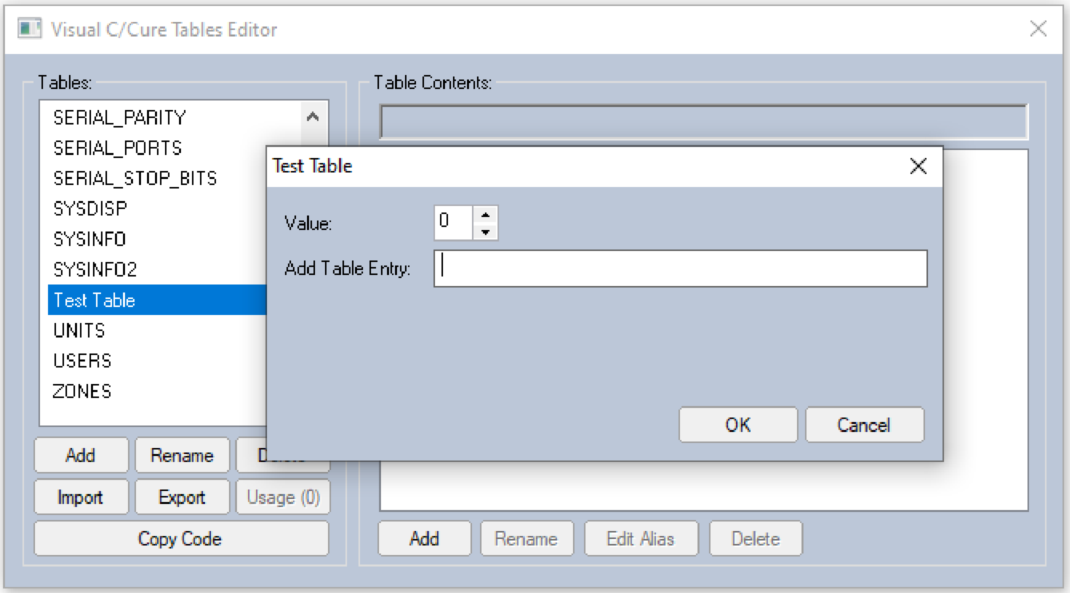
The left list shows the existing tables, along with buttons to Add, Rename and Delete tables. Note that the default system tables cannot be deleted or renamed – as such, when they are selected, the Delete and Rename buttons are greyed out.

Clicking on the Usage button lists, in the console, all the places the selected table is used.

The right list shows any table entries along, the associated integer value, and buttons to Add, Rename, Edit and Delete table entries.

To add a new table, click on the left hand “Add” button. This brings the “Add New Table” dialog when you must enter a unique table name. You also have the option to duplicate an existing table. When ready, click OK to add the new table.

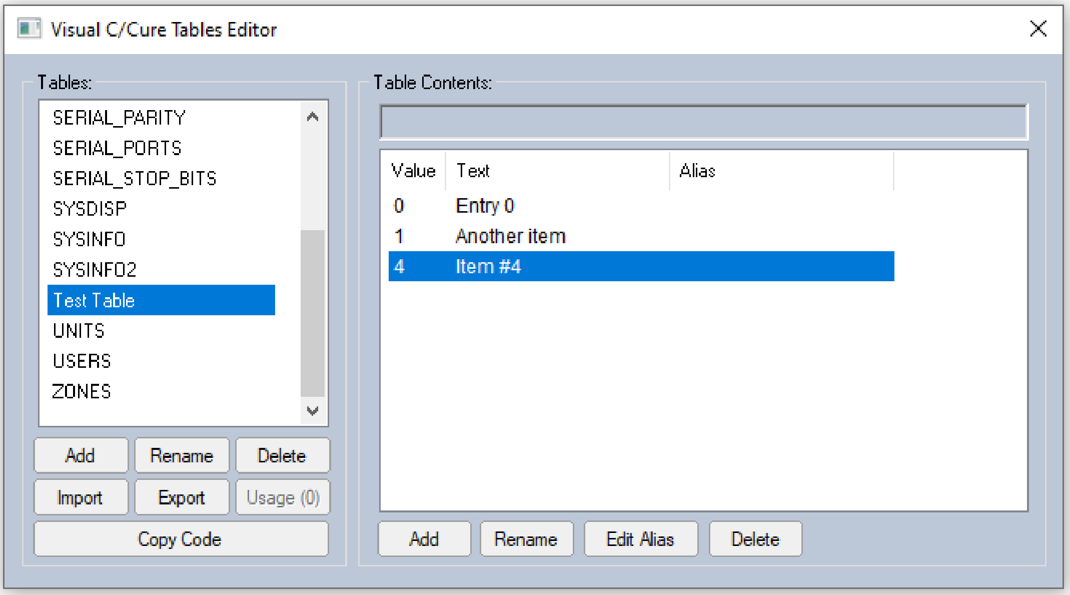
Now select the table (which will be empty) and click on the right hand “Add” button, which will bring up the table entry dialog box:-



Value: 0

Add Table Entry:

Now specify the value and string to be added to the list, and click OK. Repeat until all entries have been added.

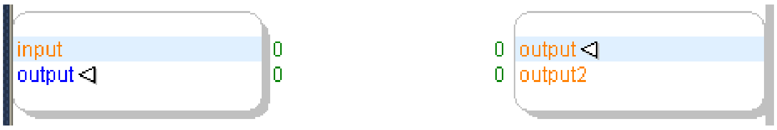


Value	Text	Alias
0	Entry 0	
1	Another item	
4	Item #4	

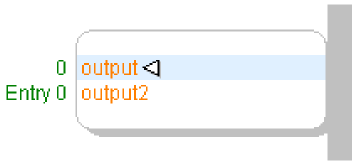
Existing entries can be changed (using the Rename button) or deleted (using the Delete button).

This table can now be used to replace any pin's data value with the relevant lookup string.

For example, if we had a container as follows:-



Click on the “output2” output pin and select “Test Table” from the Table drop down list in the Properties window. The output2 pin value of “0” will now be used to look up item “0” in the “Test Table” table, and the string value is displayed (rather than the integer value):-



If the value of output2 is changed (by double-clicking on the pin), the list of available string values (as defined by the chosen table) is shown (rather than an integer / double edit box), and the new setting can be chosen accordingly.

When values are displayed on pins with a table set, any value without an associated lookup entry will simply display the value itself.

As can be seen, there are various system tables already defined:-

ALARMGROUPS	Specifies the names of the alarm groups. See 8.6 - Alarms.
FTPFORMATS	Specifies the report upload formats available. See 11.5.5.3 - FTP.
GROUPS	Specifies the available system groups. By default, the following are defined:- **VCC_UNITSELECT** Show this pin as a “Units” source. See 6.15 - Units. **VCC_ADPSELECT** Show this pin as an “ADP” source. See 6.16 - ADP. **VCC_ALARMSELECT** Show this pin as an “Alarm” source. See 8.6 - Alarms. **VCC_COMMPORTSELECT** Show this pin as a “Comms Port” source. **TOTALS** Specifies the pin value is to be retained during a "Cold Start - Keep Totals" restart. See 11.5.9 - Reboot. **CUSTOMERKEYS** Reserved for future use. **METROLOGY** Specifies that writes to this pin can only occur if the NMI/MET hardware link is active. The pin value is also included in the Metrology checksum.
HISTORICALS	Specifies the names for the Historical Zones. See 8.8 - Historical for more information.
SERIAL_BAUD_RATES	Specifies the pre-defined baud rate values for use with comms link serial ports.
SERIAL_DATA_BITS	Specifies the pre-defined data bits values for use with comms link serial ports.
SERIAL_PARITY	Specifies the pre-defined parity settings for use with comms link serial ports.
SERIAL_PORTS	Specifies the pre-defined values for use with comms link serial ports. Using this table, along with the VCC_COMMPORTSELECT group above, allows a pin to specified as a serial comms port. Using the VCC_COMMPORTSELECT group enables the system to flag “in use” comms ports when changing port via the website and prevent multiple links attempting to use the same port.
SERIAL_STOP_BITS	Specifies the pre-defined stop bits values for use with comms link serial ports.
SYSDISP	Specifies the available options when adding a Live Menu in the Displays editor. See 8.2 - Displays for more information.
SYSINFO	Specifies the available entries in the top-level SysInfo input pin.
SYSINFO2	Specifies the available entries in the top-level SysInfo2 input pin.

UNITS	Specifies the available Units. See 6.15 - Units for more information.
USERS	Specifies the list of user names (and their associated access levels). There are only 6 levels available, with the lowest level '0' having the least restriction. Only the names themselves can be altered. See 8.2 - Displays for more information.
ZONES	Specifies the list of report zone names. See 8.5 - Reports / Live Screens for more information.

6.15 Units

The Visual C||Cure editor allows for any data pin to have an associated unit.

Although designed with S.I. units in mind, the units themselves are just another string taken from the UNITS table, and so any string values can be used.

The source of the unit value for a pin is specified via the pin's Units pop-up list in the Properties window. This pop-up list will be populated from the following sources:-

<inherit>	The pin will inherit the units from any pins linked to it.
Units table	All lookup entries from the Units table will be shown in the Units pop-up list.
VCC_UNITSELECT pins	To allow for full flexibility, the unit for a pin (call it PIN_A) can be derived via another pin (call it PIN_B). (a) Create a table (or use an existing table) containing the list of custom units to be used by PIN_A. (b) For PIN_B, select the above table in the Table pull-down list. (c) For PIN_B, check the VCC_UNITSELECT group. This will add PIN_B to the Units pop-up list on the Properties window. (d) Finally, for PIN_A select PIN_B from the Units pop-up list. The units for PIN_A will now be derived from PIN_B's lookup table, and so whenever PIN_B's value changes, its lookup string will change and thus PIN_A's unit will change accordingly. This allows the user to make dynamic units in an application – useful if the units are configurable depending on the location the device / application is to be installed.

6.16 ADP

To reduce the amount of data shown on the screen at any one time, any pin can be configured to limit the number of decimal places shown.

Note that this does not affect the calculation resolution, simply the values as they as shown on the screen.

The source of the ADP value for a pin is specified via the pin's Default ADP pull-down list in the Properties window. This pull-down list will be populated from the following sources:-

<inherit>	The pin will inherit the ADP value from any pins linked to it.
0...15	The pin will display exactly this number of digits. Values will be truncated or padded with zeros accordingly.

<variable>	The pin will display a variable number of digits after the decimal place.
VCC_ADPSELECT pins	To allow for full flexibility, the ADP setting for a pin (call it PIN_A) can be derived via another pin (call it PIN_B). (a) For PIN_B, check the VCC_ADPSELECT group. This will add PIN_B to the Default ADP pull-down list on the Properties window. (b) For PIN_A, select PIN_B from the Default ADP pull-down list. The ADP value for PIN_A will now be taken from PIN_B's value, and so whenever PIN_B's value changes, the ADP setting for PIN_A will change accordingly.

6.17 Datasets

By default, an application holds one set of default values for all persistent pins. This is called “Base” and can be seen in the Object Explorer (see 6.6 - Editor Tabs).

Visual C||Cure allow alternate sets of data to be defined, allowing the same basic application to be used for multiple situations where on the settings differ.

The user can switch between datasets in the editor by using the drop-down list in the icon bar. Doing so will update the values shown in the application.

7 C||Cure Syntax

A C||Cure codeblock can contain one or more functions, sequentially laid out. The last function will be taken as the main one and functions can call each other upwards only. For example:-

```
double foo()
{
     return 1.234;
}

int main()
{
     double d = foo();
}
```

The C||Cure language itself (as the name suggests) is closely based on the C programming language. The main differences are described below.

7.1 *Variable Declaration*

Unlike native C, all variables are initialised to zero unless an explicit initialiser is used.

Arrays can be fully initialised to a non-zero value using the following:-

```
int a[5] = { 123 };
```

In this case, all five array elements would contain the value 123.

Standard array initialisation can still be used as follows:-

```
int a[5] = { 11, 22, 33, 44, 55 };
int b[5] = { 123, 456 };              /* b[2]...b[4] initialise to zero */
```

7.2 *Function Calls*

To reduce complexity, user functions cannot be used within multi-term expressions.

For example, in standard C, the following is valid:-

```
int foo()
{
     return -1;
}

int mainfunc()
{
     double d = foo() * 2.5;
}
```

However, in C||Cure this must be split into two lines, as follows:-

```
int foo()
{
     return -1;
}

int mainfunc()
{
     double d = foo();
     d *= 2.5;
}
```

7.3 *Pointers*

The C||Cure engine does **not** support pointers. This is to ensure the code runs in a predictable “secure” environment.

However, functions can be defined with “output arguments”, specified by the addition of the '&' character, indicating a C call by reference. For example:-

```
int test(int input_num, double input_fp, double& out1, double& out2)
{
      out1 = input_num * input_fp;
      out2 = input_num / input_fp;
}
```

Using this method, a function can take multiple input parameters, perform code based on those inputs, and them set the output parameters accordingly, all without the need for pointers.

7.4 Data types

Only 32bit signed integers and 64bit doubles are supported.

Time / date values are stored as an encoded double (containing year, month, day, hour, minute, second and milliseconds).

Text strings are also supported, although in a limited fashion. A “text” variable can be defined using arrays of integers, with each integer containing 4 bytes. UTF-8 strings are supported, although note that the text length of a pin describes the number of bytes, not characters. Since the UTF-8 standard can use up-to four bytes to define a single character, care should be taken to allow enough space for the required string. See https://en.wikipedia.org/wiki/UTF-8 for more information.

7.5 Unsupported C syntax

The following elements are unsupported:-

static
struct
union
typedef
#include
#define, #if, #ifdef, etc

7.6 Standard Library Functions

Below is a list of the standard library functions provide by C||Cure:-

7.6.1 Maths Functions

double fabs(double x)	Returns the absolute value x. There is no error return.
int int(double x)	Returns the integer portion of x.
double exp(double x)	Returns the exponential value of x.
double log10(double x)	Returns the base 10 logarithm of x.
double log(double x)	Returns the base e logarithm of x.
double ln(double x)	Returns the base e logarithm of x.
double pow(double x, double y)	Returns the value of x^y.
double sqrt(double x)	Returns the square root of x.
double fmin(double x, double y)	Returns minimum value of x and y.
double fmax(double x, double y)	Returns maximum value of x and y.

int isinf(double x)	Returns -1 is x is negative infinity. Returns 1 is x is positive infinity.
int isnan(double x)	Returns 0 is x is a valid double value. Returns 1 is x is an invalid double value.
double pi()	Returns the value of π.
double sin(double x)	Returns the sine of x.
double cos(double x)	Returns the cosine of x.
double tan(double x)	Returns the tangent of x.
double asin(double x)	Returns the arc sine of x.
double acos(double x)	Returns the arc cosine of x.
double atan(double x)	Returns the arc tangent of x.
double atan2(double y, double x)	Returns the arc tangent of y/x.
double sinh(double x)	Returns the hyperbolic sine of x.
double cosh(double x)	Returns the hyperbolic cosine of x.
double tanh(double x)	Returns the hyperbolic tangent of x.
int atoi(int[] x)	Converts the string stored in int array x to an integer.
double atof(int[] x)	Converts the string stored in int array x to a double.

7.6.2 Time / Date Functions

int nftimeyear(double dTime)	Decodes and returns the year from a double encoded time. The full year is returned, e.g. 2007.
int nftimemonth(double dTime)	Decodes and returns the month from a double encoded time. January = 1, December = 12.
int nftimeday(double dTime)	Decodes and returns the day from a double encoded time. The value will be between 1 and 31.
int nftimehour(double dTime)	Decodes and returns the hour from a double encoded time. The value will be between 0 and 23.
int nftimeminute(double dTime)	Decodes and returns the minute from a double encoded time. The value will be between 0 and 59.
int nftimesecond(double dTime)	Decodes and returns the second from a double encoded time. The value will be between 0 and 59.
int nftimems(double dTime)	Decodes and returns the milliseconds from a double encoded time. The value will be between 0 and 999.
double nfbuildtime(int nYear, int nMonth, int nDay, int nHour, int nMin, int nSec)	Creates and returns an encoded double time and date as specified in the arguments.

int nfsettime(int nYear, int nMonth, int nDay, int nHour, int nMin, int nSec)	Sets the system time as specified by the arguments. Returns 1 (TRUE) or 0 (FALSE) depending on whether a valid time was specified. Example, to set the time to 1st December 2021 10:30:00:- `nfsettime(2021, 12, 1, 10, 30, 0);`

7.6.3 String Functions

int nfgettableentry(string x, string y) int nfgettableentry(int[] x, int[] y)	Attempts to find the index of string y in table x. If table x does not exist in the application, or string y is not found within the table, the function returns -1.
int nfstrcmp(int[] x, int[] y)	nfstrcmp behaves as per the standard C strcmp() function, but takes int array strings.
int nfstrcasecmp(int[] x, int[] y)	nfstrcasecmp behaves as per the standard C strcasecmp() function, but takes int array strings.
int[] nfsprintf("format", …)	nfsprintf aims to follow all the usual rules of the C printf(), but will return the resulting string as an int array.

7.6.4 Debug Functions

int printf("format", …)	printf will send output to the console in Visual C||Cure. It aims to follow all the usual rules of the C printf(). Known differences are:- • A '\n' is appended to every printf if one isn't already present. Returns the number of characters printed.
int nfevent("format", …)	nfevent behaves a printf() above, but will store the resulting string in the Application Event Log. Returns the number of characters printed.

7.6.5 Licensing Functions

int nflcisvalid(int x, int y)	Check if a license is currently valid. x (currently unused) denotes the license index. If y is non-zero, then the license manager will consider a license to be valid even if the current license count is <= 0. Returns 1 if a license was valid. Otherwise returns 0.
int nflcuse(int x, int y)	Consume a license. x (currently unused) denotes the license index. If y is non-zero, then the license manager will allow a license to be released even if the current license count is <= 0. Returns 1 if a license was consumed. Otherwise returns 0.

int nflcset(int[] x, int[] y)	Add/set licensing information. String x (containing an encrypted license key) is decrypted using string y as the key. The result data is passed to the internal license manager for decoding. Always returns 0.

8 Information Tabs

The left hand window of the Visual C||Cure Editor contains a number of information / configuration tabs, through which many ancillary functions of an application can be configured.

Each tabs can be moved or hidden to suit the user's needs. Tabs can be shown / hidden via View | Bars and then selecting / deselecting the tabs as required.

8.1 Workspace

The Workspace tab shows a hierarchical tree of all the containers and codeblocks within the application.

Any password protected codeblocks or containers will not show their hierarchical contents unless the default Visual C||Cure password (see 6.1 - System Preferences) matches the password of the relevant item.

Below is a list of the information columns:-

Block	Shows the name of the container / codeblock.
Status	Indicates the status of the block:- **ok** this block is not in the library. **Code Error** this codeblock has a syntax error. **No Library File** the library block for this item is missing. **Sync** this block is in sync with the library. **Out of Sync** a newer version of this block is available in the library.
Time	Shows the cycle time (in milliseconds) for the block. This item only updates when running in Remote Debug mode (see 12 - Remote Debug).
Inst's (Instructions)	Shows the number of instructions executed in the block. This item only updates when running in Run mode or Remote Debug mode (see 12 - Remote Debug).
Iter's (Iterations)	Shows the number of iterations (if any) this block performed. Codeblocks do not support iterations and will be greyed out. This item only updates when running in Run mode or Remote Debug mode (see 12 - Remote Debug).

8.2 Displays

The Displays tab allows the user to setup or edit the hierarchical menus and display items that will be available via a "remote" link (e.g. web server or XML comms), "local" HMI panel or “lid” display.

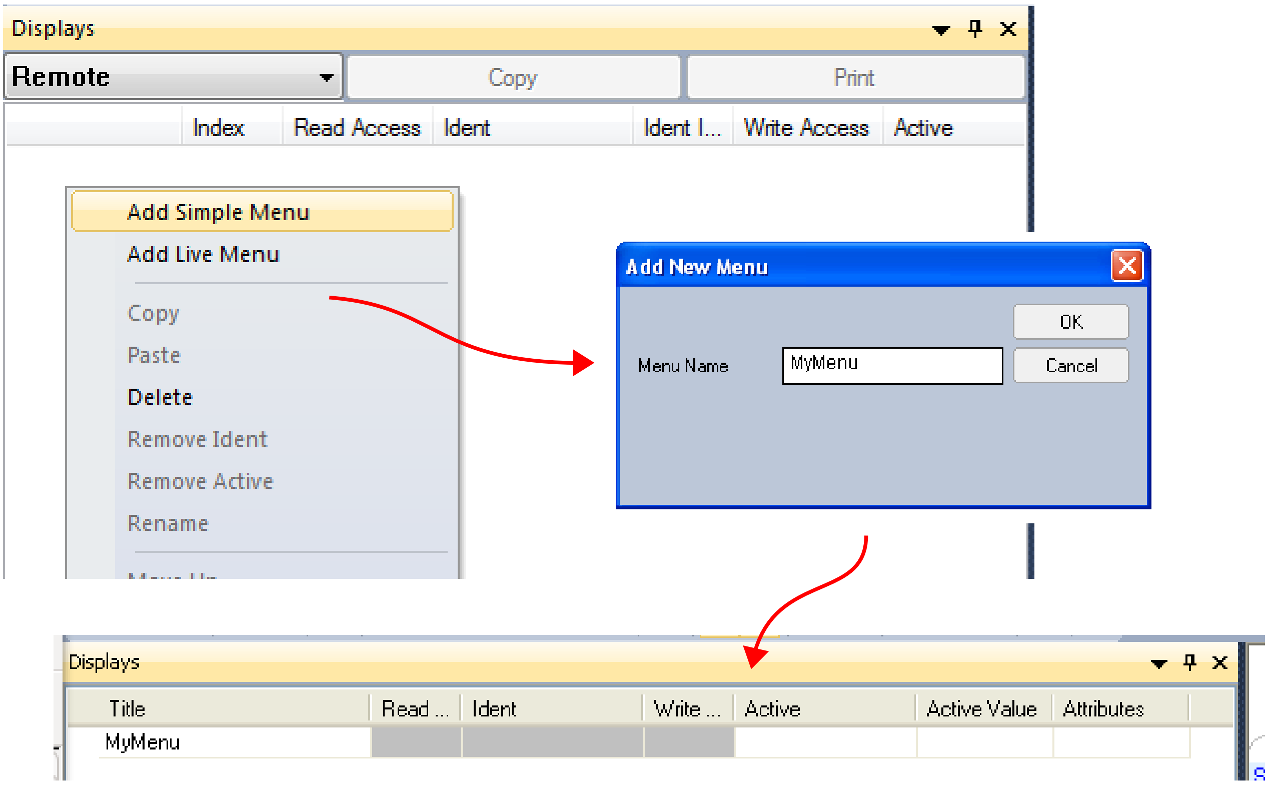
Note: for detailed information on each particular display link, refer to the Visual C||CURE Comms Links document (reference NFVCCL).

The following features are supported:-

Simple menus	Simple menus are the standard type of menu, allowing sub-menus or display items, but not a mix of the two.
Live menus	Live menus enable a page of “live” data to be displayed, based on an existing Report layout (see 8.5 - Reports / Live Screens). Live menus cannot contain sub-menus.

Read / Write user access control	Any display item can be configured to restrict both read and write access to a particular access level (or above).
Control / Status items	Display items can be configured to show the value of a "status" data pin when reading, but send changes to an alternative "control" data pin when writing.
Active / Inactive menus	All menus and display items can be setup with an "active" trigger pin (and value). When specified, the menu / display item will only be visible when the relevant active pin matches the configured active value. If no active pin is specified, the menu / display item will always be visible.

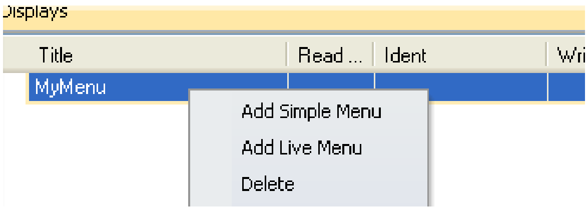
To add a new top-level menu, select the relevant "remote" or "local" display tree from the drop-down menu. Then right-click on the empty space in the tab and select Add Simple Menu. Enter the menu name and click OK.

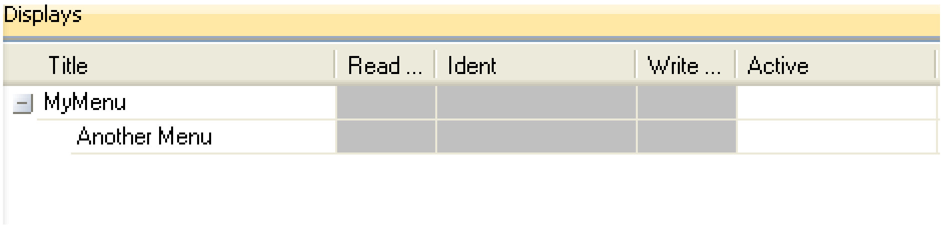


Menu Name: MyMenu

Title	Read ...	Ident	Write ...	Active	Active Value	Attributes
MyMenu						

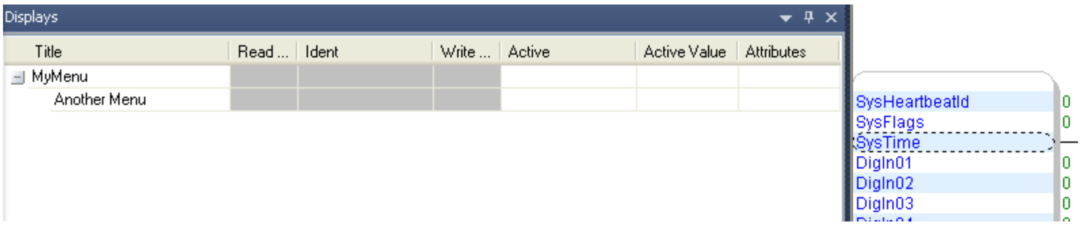
To add a sub-menu, right-click on an existing menu entry and select Add Simple Menu. Enter the menu name and click OK.





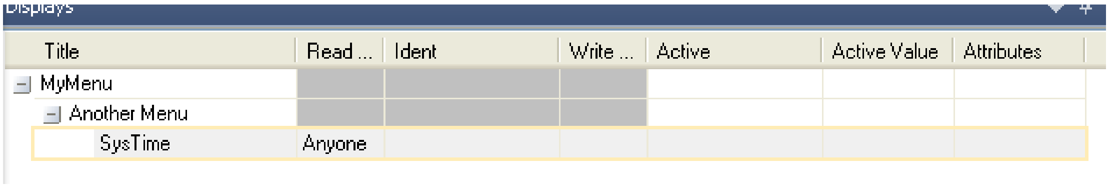
Title	Read ...	Ident	Write ...	Active
MyMenu				
Another Menu				

To add a display item to a menu, navigate within the application editor window to the relevant application container or codeblock that has the required pin (or locate the pin in the Object Explorer window), and drag the pin onto the relevant menu item.



Title	Read ...	Ident	Write ...	Active	Active Value	Attributes
MyMenu						
Another Menu						

This will insert the selected data pin into the target menu.



Title	Read ...	Ident	Write ...	Active	Active Value	Attributes
MyMenu						
Another Menu						
SysTime	Anyone					

If an array pin is drag-and-dropped, the user will get the option to add each array item to the display tree. If only a single item is chosen, the item's Index spinner can be used to select the required item in the array.

All menus and display items can be moved up and down within the hierarchy. Right-clicking on an item will show the options:-

Move Up	(shortcut 'u')	Moves the item up one slot
Move Down	(shortcut 'd')	Moves the item down one slot
Move Parent	(shortcut 'l')	Moves the item left / up one hierarchical level
Move Child	(shortcut 'r')	Moves the item right / down one hierarchical level

If multiple items are selected within the same menu, they can be moved up/down together.

Items can be deleted using right-click Delete. Note that deleting a menu will also delete any of its children.

Data items can be "located" within the application via right-click Locate or by double-clicking the display item.

Below is a list of the information columns:-

Title	Shows the name of the menu / display item. Menu names can be changed by clicking on the title cell and entering new text as required. Display item names are either the pin name or base pin alias (if set).
Index	For array pins, this specifies the array index to be used when reading data. This will be greyed out for non-array pins.
Read Access	This defaults to level 5 (Read Only), so any user with access level 5 (or less) will be able to see this item. The access level can be changed via the drop-down list. The access levels are defined using the USERS table (see 6.14 - Tables).
Ident	For user-configurable input pins, the Ident sets the pin to be written when changing data. By default, the Ident is the same as the original pin, but this can be changed by dragging another pin onto the Ident cell. The Ident can also be removed (so that the display item cannot be written to) by using right-click Remove Ident.

Ident Index	For array pins, this specifies the array index to be used when writing data. This will be greyed out for non-array pins.
Write Access	This defaults to level 1 (Admin), so any user with access level 1 (or less) will be able to change this item. The access level can be changed via the drop-down list. The access levels are defined using the USERS table (see 6.14 - Tables).
Active	Dragging a pin onto the Active cell sets that pin as the active trigger. When used, the pin will only be shown on the display if the pin's current value matches the active value (see below). The Active pin can be removed by right-click Clear Active.
Active Value	If an active pin has been configured, this sets the integer value at which the display item becomes “active”. The active value can be changed using the up / down spinner buttons.
Attributes	This is a general purpose string field that allows the user to assign each display item its own custom “attribute” string. See 8.2.1 - Display Attributes for more information on currently supported system attributes. The attribute string is also transmitted when using XML comms <Displays> and <Live_Displays> requests.

8.2.1 Display Attributes

Each entry in a display tree can contain its own attributes string. This can be entirely custom to the application writer and can be accessed via the XML comms link.

Alternatively, there are several "reserved" keywords that are used by the local panel and/or the web server to provide certain enhanced features.

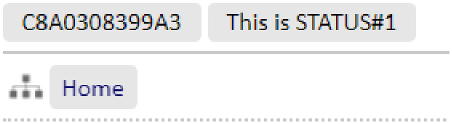
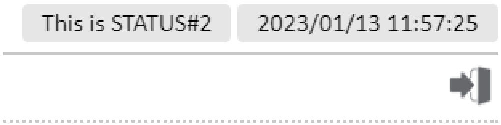
The web server contains the ability to show a mimic of the local panel, and will attempt to recreate the layout of the local panel as best it can.

The table below lists the relevant attributes, along with a description and whether the attribute is supported by the web server, local panel or lid.

8.2.1.1 Menu Items

<table>
<tr><th>Attribute</th><th>Description</th><th>Web Server</th><th>Local Panel</th><th>Lid</th></tr>
<tr><td>size</td><td>Specifies the layout of the screen by setting the number of rows and columns, using the format "size=<rows>x<cols>".
Panels are laid out / numbered left-to-right, then top-to-bottom.
For example, "size=4x6" sets the screen to 4 columns and 6 rows, with the individual panels laid out as:-
<table>
<tr><td>1</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td>5</td><td>6</td><td>7</td><td>8</td></tr>
<tr><td>9</td><td>10</td><td>11</td><td>12</td></tr>
<tr><td>13</td><td>14</td><td>15</td><td>16</td></tr>
<tr><td>17</td><td>18</td><td>19</td><td>20</td></tr>
<tr><td>21</td><td>22</td><td>23</td><td>24</td></tr>
</table>
If no size attribute is specified, the display defaults to 2 columns and 4 rows.</td><td>Y</td><td>Y</td><td>N</td></tr>
<tr><td>fs</td><td>Sets the default font sizes (big and small for Title, Value and Units), using the format "fs=<TB>,<TS>,<VB>,<VS>,<UB>,<US>" where:-
<TB> Title big font size (default = 26)
<TS> Title small font size (default = 14)
<VB> Value big font size (default = 26)
<VS> Value small font size (default = 20)
<UB> Units big font size (default = 26)
<US> Units small font size (default = 14)
Any missing values will fallback to the system defaults.
For example "fs=16,10,,,18,14" sets the Title and Units font sizes, but leaves the Value font sizes as the defaults.
Another example is "fs=18" which just sets the the Title big font size and leaves the all other font sizes as their defaults.
If no fs attribute is specified, all font size use their defaults.
Note that these font sizes can be overridden for individual items using the "st" parameter below.</td><td>Y</td><td>Y</td><td>Y</td></tr>
</table>

bg	Sets the background colour for the page using a 6 digit hex RGB value. For example "`bg=ff0000`" sets the background to red. The default is white (`ffffff`).	Y	Y	Y
cols	By default, the screen contains only a single column of data items. This option can be used to set the number of columns and their percentage widths. The widths are set using a list of comma separated percentages. The number of percentages given sets the number of columns to create. A maximum of 10 columns can be specified. For example "`cols=25,50,25`" creates three columns, 25% wide, 50% wide and 25% wide.	Y	N	N
clog	Specifies how to show/hide this item on the Constants Log report. ha (H)ide (A)lways hi (H)ide when item (I)nactive hc (H)ide when item's (C)hildren inactive	Y	N	N
chart	Defines this menu to be displayed as a chart on the lid display. The first four database items are used to determine the chart appearance and values. Item 1 Used for the value of the chart. This item’s pin name is displayed one the first line of the lid display, with the chart shown beneath it. Item 2 Used for set the minimum value shown on the chart. This item is optional and, if missing, the minimum will default to 0. Item 3 Used for set the maximum value shown on the chart. This item is optional and, if missing, the maximum will default to 100. Item 4 Used for set the chart update time in application cycles (each cycle being 500ms). For example, a setting of 1 will update the chart every ½ second. A setting of 120 will update the chart every minute. This item is optional and, if missing, will default to 1. The chart automatically scales the value to fit on the display, with the minimum value at the bottom and the maximum value at the top. If the minimum or maximum values are changed, the chart will automatically redraw itself to adjust to the new min/max values. If the current chart value lies outside the min/max range, the letters “OR” (‘O’utside ‘R’ange) will be displayed at the top-right corner. Note that alarms are not displayed on the lid when the chart is in use.	N	N	Y

appstatus	Defines this menu to be displayed in the header.  The first “active” item within the menu is displayed on the left side of the header (to the right of the hostname).	Y	N	N
appstatus2	Defines this menu to be displayed in the header.  The first “active” item within the menu is displayed on the right side of the header (to the left of the datetime).	Y	N	N

8.2.1.2 Database Items

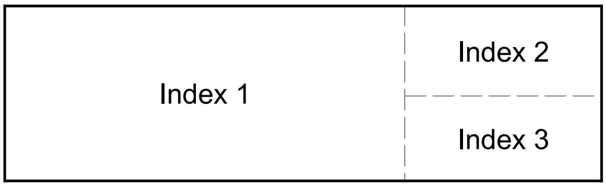
Attribute	Description	Web Server	Local Panel	Lid
id	Sets the location and type of panel used to display the database item contents, using the format `"id=<start>,<end><options>"`.			
	The `<start>` value specifies the start panel to use.	Y	Y	N
	The <end> value can be used to specify the end panel, which allows multiple rows and/or columns to be spanned.	Y	Y	N
	Both values are optional. If no value is specified, the next sequential panel is used as the start position, and if only 1 value is given, this sets the start position and no row/column spanning occurs.			
	For example, using the 4x6 layout above, specifying `"id=1,21"` will create a single panel covering positions 1, 5, 9, 13, 17 and 21. Whereas specifying `"id=10,15"` will create a single panel covering the middle 4 positions 10, 11, 14 and 15.			
	On the Local Panel, the "whole panel" can be specified using `"id=*"`.	N	Y	N
	The `<options>` section allows the "type" of panel to be specified.			
	b Create an "empty" edit panel … no title, value or units	N	Y	N
	e Create an "empty" panel	Y	Y	N
	t Show the item title (including array field names)	Y	Y	Y
	n Show the item name (item name only, no array field names)	Y	Y	Y
	i Show the item field (array field name only, no item name)	Y	Y	Y
	v Show the item value	Y	Y	Y
	u Show the item units	Y	Y	Y
	a Create an "auto-edit" screen - when the screen loads, the relevant data entry keyboard is displayed and the user is prompted to enter data	Y	Y	N
	f When editing, force a value to be entered (i.e. no zero or empty string)	Y	Y	N
	p Show the item as a password (shows ‘*’ instead)	Y	N	N
	px Enter hidden pin code ('x' hidden digits - shows '*' instead)	N	Y	N
	Px Enter visible pin code ('x' visible digits)	N	Y	N

	T Set the "toggle accept" item when showing a toggle pin (see the 'tg' parameter below) If no options are given, the panel defaults to 'tvu'.	Y	Y	N
fg	Sets the foreground colour for Title, Value and Units text, using 6 digit hex RGB values. A choice of 1, 2 or 3 values can be specified, as follows:- `fg=<c>` Use colour `<c>` for Title, Value and Units `fg=<c1>,<c2>` Use colour `<c1>` for Title and Units text Use colour `<c2>` for Value text `fg=<c1>,<c2>,<c3>` Use colour `<c1>` for Title text Use colour `<c2>` for Value text Use colour `<c3>` for Units text The default is black (`000000`).	Y	Y	N
bg	Sets the background colour for the panel using a 6 digit hex RGB value. For example `"bg=ff0000"` sets the background to red. The default is light grey (`d3d3d3`).	Y	Y	Y
st	Sets the style of the Title, Value and Units text, using the format `"st=<ts><to>,<vs><vo>,<us><uo>"`. `<ts>` Sets the font size for the Title text `<to>` Sets the options for the Title text `<vs>` Sets the font size for the Value text `<vo>` Sets the options for the Title text `<us>` Sets the font size for the Units text `<uo>` Sets the options for the Title text The available options are as follows:- `b` Show in bold `c` Centre text `i` Show in italics `s` Show with strike-through `u` Show with underline The size and options sections are optional for all 3 settings. For example `"st=26bc"` sets the Title text to 26pt, bold and centred and leaves the Value text and Units text as default. Using `"st=26bc,,u"` sets the Title text to 26pt, bold and centred, leaves the Value text as default and adds underlining to the Units text.	Y	Y	N
tg	A group of 3 items can be combined to create a single "toggle" pin. Each single toggle pin has 3 slots, and a layout "style" is specified to determine the look and feel of the toggle pin. When displayed on the local panel, all 3 items are shown in a single panel, and can be toggled on/off by pressing the panel. The format is `"tg=<style><slot><options>,<bit>,<bg>,<fg>"`. `<style>` Layout style `<slot>` Slot number with the layout `<options>` What text to display:- `t` Show pin Title `v` Show pin Value	Y	Y	N

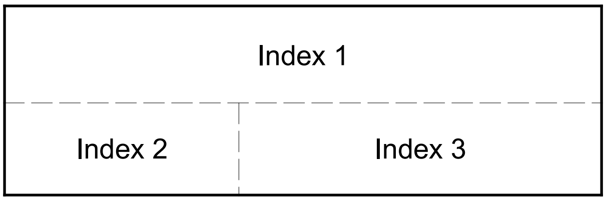
	u	Show pin Units
	vu	Show pin Value and Units
<bit>	Bit number to toggle	
<bg>	Panel background colour when the toggle is "on"	
<fg>	Text foreground colour when the toggle is "on"	

Two styles are currently supported:-

a Index 1 : Full height, 70% width
Index 2 : 50% height, 30% width
Index 3 : 50% height, 30% width



b Index 1 : 50% height, full width
Index 2 : 50% height, 40% width
Index 3 : 50% height, 60% width



The default colours are:-

	Toggle "off"	Toggle "on"
Background	cccccc (light grey)	00ff00 (green)
Foreground	808080 (grey)	000000 (black)

The <bit>, <bg> and <fg> settings should only be set on the first item in the group.

tg
(cont)

For example, consider the following item attributes:-

[item1] `id=3 tg=b1v,0,ff0000,808080`

- Use style 'b', slot 1
- Show the pin Value
- Set/clear toggle bit 0
- Use a toggle "on" background colour of `ff0000` (red)
- Use a toggle "on" foreground colour of `808080` (grey)

[item2] `id=3 tg=b2t`

- Use style 'b', slot 2
- Show the pin Title

[item3] `id=3 tg=b3t`

- Use style 'b', slot 3
- Show the pin Title

[item4] `id=4 tg=b1v,1,ff0000,808080`

- Start next toggle item ...

When the toggle "accept" item is selected (see the "`id`" parameter above), the on/off value of bit 0 will be written to the ident pin for item 1.

<table>
<tr><td></td><td>There is also the option of just using the combined layout, but not the "toggle" feature. To achieve this, simply omit the toggle bit values.</td><td></td><td></td><td></td></tr>
<tr><td>sort</td><td>Sets the sort type when editing database items with an associated table list.
If no sort option is specified, the list of choices is presented in the order of increasing index values, but the user can also be shown the list sorted is various ways.
Take, for example, the following table:-
<table>
<tr><th>Value</th><th>Text</th></tr>
<tr><td>-1</td><td>First entry</td></tr>
<tr><td>3</td><td>Another entry in here</td></tr>
<tr><td>99</td><td>Invalid choice</td></tr>
</table>
This can be displayed using the following sort options:-
<table>
<tr><td><no sort option specified></td><td>Default table order:-
• First Entry
• Another entry in here
• Invalid choice</td></tr>
<tr><td><code>sort=re</code></td><td>Reverse table order:-
• Invalid choice
• Another entry in here
• First Entry</td></tr>
<tr><td><code>sort=az</code></td><td>Alphabetical order:-
• Another entry in here
• First Entry
• Invalid choice</td></tr>
<tr><td><code>sort=za</code></td><td>Reverse alphabetical order:-
• Invalid choice
• First Entry
• Another entry in here</td></tr>
</table></td><td>Y</td><td>N</td><td>N</td></tr>
<tr><td>nolog</td><td>When used with writeable items, specifies that no Event Log entry should be stored.</td><td>Y</td><td>Y</td><td>N</td></tr>
<tr><td>span</td><td>Sets the number of columns this item should span.
Only applicable if the "cols" attribute has been set for this screen.</td><td>Y</td><td>N</td><td>N</td></tr>
<tr><td>date</td><td>Specifies that this item is to be shown as a date, according to the current system date format.</td><td>Y</td><td>N</td><td>N</td></tr>
<tr><td>report</td><td>Allows the contents of a report to be displayed on the local panel.
The report is specified using the format <code>"report='<report_name>'"</code>.
If the specified report is a "live screen", the report will be generated using a snapshot of the current live database. Otherwise the most recent matching report is extract from flash storage and used to generate the report text.
On the local panel itself, when the relevant panel is pressed, the report text is displayed, along with up/down scroll buttons, allowing the operator to view the entire report. An exit button allows the operator to hide the report.
Note that this assumes the item has been correctly configured as a</td><td>Y</td><td>Y</td><td>N</td></tr>
</table>

	button.			
qrcode	Used to display a QR code in the panel, using the format `"qrcode='<report_name>'"`, where <report_name> sets the name of the report to use to generate the QR code contents. As per the "report" parameter above, if the specified report is a "live screen", the report will be generated using a snapshot of the current live database. Otherwise the most recent matching report is extract from flash storage and used to generate the report text.	Y	Y	N
ppb	The size of a QR code depends on the quantity of text held within the code. More text results in a bigger QR code. Depending on the configured report, the pixels-per-bit (ppb) can be set to increase/decrease the size of the QR code image to allow it to fit within the available panel size. For example, `"ppb=2"` creates the QR code using 2 pixels-per-bit. This parameter is optional and the default setting is 4.	Y	Y	N
list	Creates a selection list for this item, with one list item for each of the valid table entries for this database item. Long lists are supported via Up/Down buttons. The attribute format is:- `"list=<items>,<bg_unsel>,<fg_unsel>,<bg_sel>,<fg_sel>"` items number of items to show per page bg_unsel background colour when item not selected fg_unsel foreground colour when item not selected bg_sel background colour when item is selected fg_sel foreground colour when item is selected For example `"list=5,0000ff,ffffff,ff0000,00ff00"` creates a list with 5 items per page, with a Blue background / White foreground when not selected, and a Red background / Green foreground when selected.	N	Y	N
alarms	Flags that this item, when pressed, will display the Alarms screen. Note that this only works on the Local Panel … on the Website, you can just press the existing "Alarms" button on the web page header.	N	Y	N

8.3 Communications

Communications forms an essential part of any embedded system.

The Visual C||Cure editor provides a powerful comms configuration system that is able to support any number and type of comms links. The comms links themselves are highly configurable, supporting multiple polls per link, and multiple data items per poll.

Please refer to the separate Visual C||CURE Comms Links document (reference NFVCCL) for a detailed explanation on creating and configuring the various available comms links.

8.4 Datamaps

The datamaps window allows the user to map comms link items to database pins.

Please refer to the separate Visual C||CURE Comms Links document (reference NFVCCL) for a detailed explanation on creating and configuring the various available comms links.

8.5 Reports / Live Screens

A customisable report/screens editor is provided, allowing any number of data pins to be stored / viewed together in a single page of information.

Any number of reports and live screens can be added to an application.

Reports can be triggered to be stored using the output of any data pin within the application. They are kept on the target using non-volatile storage, and can be retrieved from a remote target via a suitable comms link or via Remote Debug.

Different "zones" are provided to allow for more / less frequent reports.

16 report zones are available, each of which is designed to provide a maximum number of reports being stored at a certain frequency, as follows:-

Zones	Suggested Usage	# Slots	Maximum Average Trigger Rate
Large 1...6	Yearly reports	1500	Once every minute.
Small 1...10	Monthly reports	250	Once every 5 minutes.
98	Print Only	N/A	Used for sending instant reports to a printer.
99	Snapshot reports	1	This is a temporary report stored in RAM. No maximum trigger rate applies.

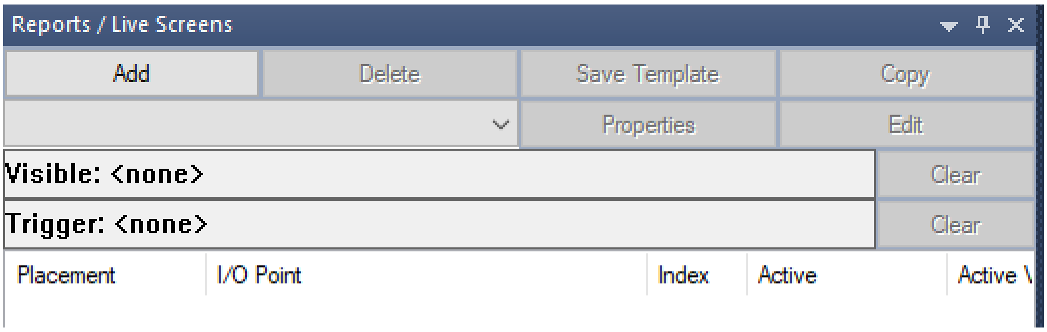
The Maximum Average Trigger Rate must be respected to guarantee a 25 year life for the target hardware in worst case conditions.

Caution : FLASH memory technology is used to store the report data. This technology has a limited "lifetime", so to prevent damaging the FLASH memory, each report slot should not be written **on average** more than as specified in the above table. Hardware faults due to over-use of the FLASH memory will invalidate any warranty. Usage counters are stored in non-volatile memory to allow monitoring over the lifetime of the device.

Live screens (unlike reports) are not kept historically on the target. As the name suggests, they are for viewing "live" data and must be added to the displays tree (see 8.2 - Displays for more information).

Note that, since they can only be used for viewing data, any user-configurable input pins cannot be changed using a live screen.

The Reports / Live Screens window allows reports to be selected from a drop-down list, added, removed, edited and saved to the library.



Visible: <none>

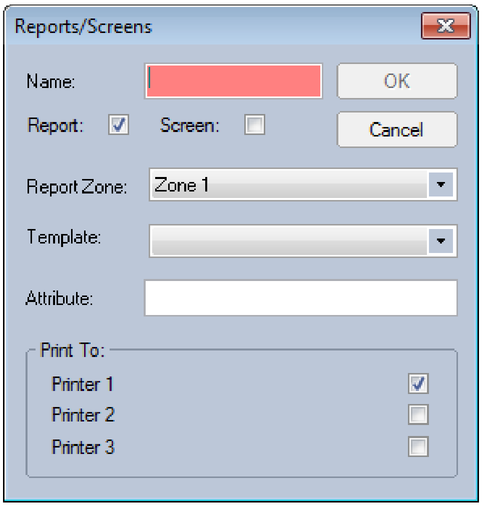
Trigger: <none>

The following controls are shown:-

Add	Adds a new report / live screen.
Delete	Deletes the selected report / live screen.
Save Template	Saves the selected report / live screen to the library as a template. All saved reports are listed in the Template drop-down list on the Reports/Screens properties dialog.

Copy	This copies the report layout to the clipboard in HTML format, useful for documentation purposes.
Properties	Shows the Reports/Screens properties dialog for the selected report.
Edit	Opens the report editor tab for the selected report.
Visible	Shows the data pin to be used to control if this report is visible via the website. The trigger is set by dragging an application data pin on the control. Whenever this data pin is non-zero, the report will be stored as per the report properties.
Trigger	Shows the data pin to be used as the report trigger. The trigger is set as per the Visible control.

To create a new report, click on the Add button, which brings up the Reports/Screens properties dialog.



Reports/Screens

Name:

Report: ☑ Screen: □

Report Zone: Zone 1

Template:

Attribute:

Print To:

Printer 1 ☑

Printer 2 □

Printer 3 □

Below is an explanation of the available options:-

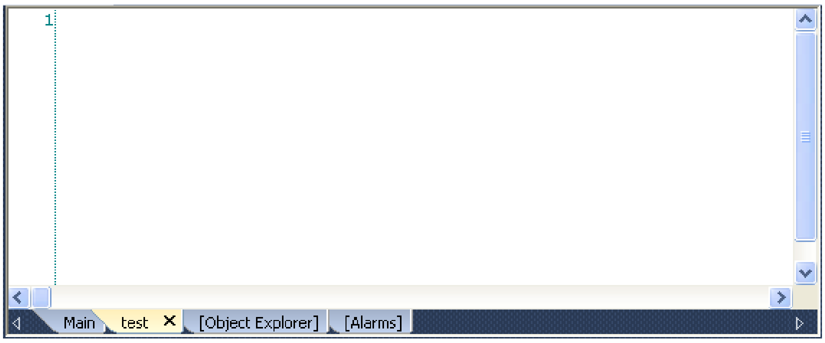
Name	Enter unique report / live screen name. This is also how live screens are referred to on the displays tree and website, and how reports are listed under the System Logs web page.
Report	Selects whether this can be used a report.
Screen	Selects whether this can be used a live screen.
Report Zone	Selects the report zone to use.
Template	Selects the library template (if any) to use for this report.
Attribute	Sets the report attribute(s) for this report. See below.
Print To	Sets the default printer(s) that this report should be sent to whenever generated.

Report attributes

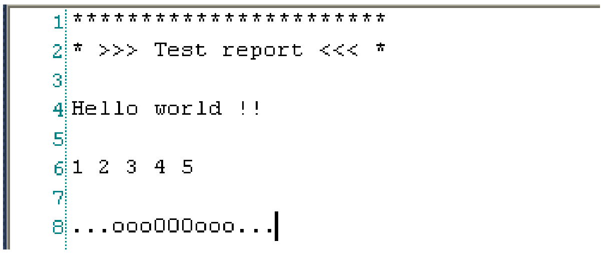
<table>
<tr><td>dl</td><td>List of download formats for the report.
One or more of the following can be set:-
<b>text</b> Plain text format
<b>tsv</b> Tab Separated Value format
<b>tfx</b> FlowCal TFX format
<b>pdf</b> PDF format
The default type is "text".</td></tr>
<tr><td>file</td><td>Specify the filename format when the report is uploaded via FTP.
The various format options are detailed in the FTP Upload Directory section (see 11.5.5.3 - FTP).
If not defined, it will default to %r-%D%T.<ext> where <ext> is the extension type for the report being generated (e.g. tsv, txt).</td></tr>
<tr><td>fs</td><td>Set the font size when printing the report to Postscript printers.
The following table shows approximately (depending on the printer) how many rows and columns can be printed at various font sizes:-<table>
<tr><th>Font Size</th><th>Rows</th><th>Columns</th></tr>
<tr><td>6</td><td>100</td><td>144</td></tr>
<tr><td>8</td><td>78</td><td>108</td></tr>
<tr><td>10</td><td>64</td><td>86</td></tr>
<tr><td>12</td><td>54</td><td>72</td></tr>
<tr><td>14</td><td>46</td><td>62</td></tr>
</table>The default font size is 10.</td></tr>
<tr><td>type</td><td>Sets the (optional) report type.
<b>fctfx</b> Specify the report can be extracted in FlowCal TFX format via the XML comms link.
<b>qrcode</b> Specify the report should be shown as a QR code.</td></tr>
</table>

ident	This attribute can be used to show additional information about a report when being shown in a list of reports on the website or via the xmlcomms link.
	By default, the report name and date-stamp are normally displayed but the value of one of the report placements can also be shown to help identify a particular report.
	The attribute consists of three sections, denoting prefix, value and suffix as follows:-
	`ident='<prefix>','<placement>','<suffix>'`
	The `<prefix>` section specifies the text to be added before the ident value.
	The `<placement>` section specifies the report placement name from which to take the value to be displayed.
	The `<suffix>` section specifies the suffix text to be added after the ident value.
	Note that all sections must be present but can be empty.
	For example, if the report (called "Prove Report") contained a placement called Meter Run, you should use the attribute:-
	`ident='-MR','Meter Run',''`
	If the report meter run value was 7 then this would produce a report listing entry of:-
	`2019/08/12 15:44:54 (Prove Report-MR7)`

Once the required options have been selected, click OK. This will then open the report editor and show a blank report:-

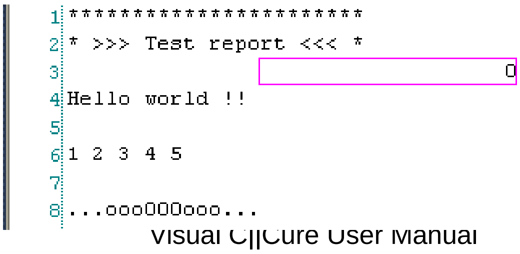


Background text can be added to suit the needs of the user.



```
***********************
* >>> Test report <<< *

Hello world !!

1 2 3 4 5

...oooOOOooo...
```

Next, a “placement” needs to be added for each item of data that is to appear on the report. Right-click in the report editor, select New Placement and a new placement box will appear under the cursor.



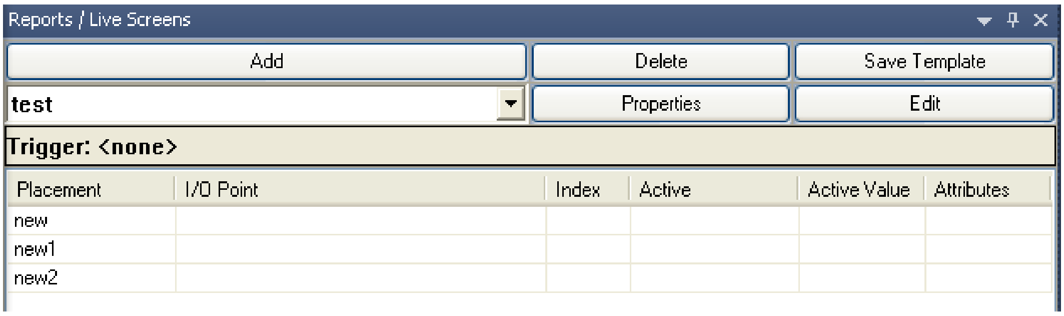
```
***********************
* >>> Test report <<< *
                                    0
Hello world !!

1 2 3 4 5

...oooOOOooo...
```

This box can be dragged around the report editor screen to the required position.

The following placement properties can also be changed via the Properties window:-

Name	Sets the name of the placement.
ADP	Allows the “After Decimal Place” value to be overridden. Setting to <inherit> takes the ADP value from the application pin property. See 6.16 - ADP for more information.
Width	Sets the number of characters shown for the placement.
Leading Zeros	Specifies whether to pad to the field width with leading zeros.
Left Align	Specifies whether to align the data with the left edge of the placement.
Bold	Specifies whether to show the value in bold text.
Underline	Specifies whether to show the value in underlined text.
Date Format	Specifies if, and how, to format this value as a date. Note that this setting only applies if the application data pin has been configured as a Date/Time pin (see 6.12 - Pin Properties). If left blank, the default format will be the target default (as set via the web server or suitable comms link). The following default formats are available:- %Y/%m/%d %H:%M:%S %Y/%m/%d %H:%M %Y/%m/%d %Y/%d/%m %H:%M:%S %Y/%d/%m %H:%M %Y/%d/%m %H:%M:%S %H:%M Custom formats can be created using the following:- %a Abbreviated weekday name %A Full weekday name %b Abbreviated month name %B Full month name %c Date and time representation appropriate for locale %d Day of month as decimal number (01-31) %H Hour in 24-hour format (00-23) %I Hour in 12-hour format (01-12) %j Day of year as decimal number (001-366) %m Month as decimal number (01-12) %M Minute as decimal number (00-59) %p Current locale's A.M./P.M. indicator for 12-hour clock %S Second as decimal number (00-59) %U Week of year as decimal number, with Sunday as first day of week (00–53) %w Weekday as decimal number (0-6; Sunday is 0) %W Week of year as decimal number, with Monday as first day of week (00–53) %x Date representation for current locale %X Time representation for current locale %y Year without century, as decimal number (00-99) %Y Year with century, as decimal number %z, %Z Either the time-zone name or time zone abbreviation; no characters if time zone is unknown %% Percent sign

Multiple placement points can be added and positioned as required, and each placement on the report is listed in the Report / Live Screens window.

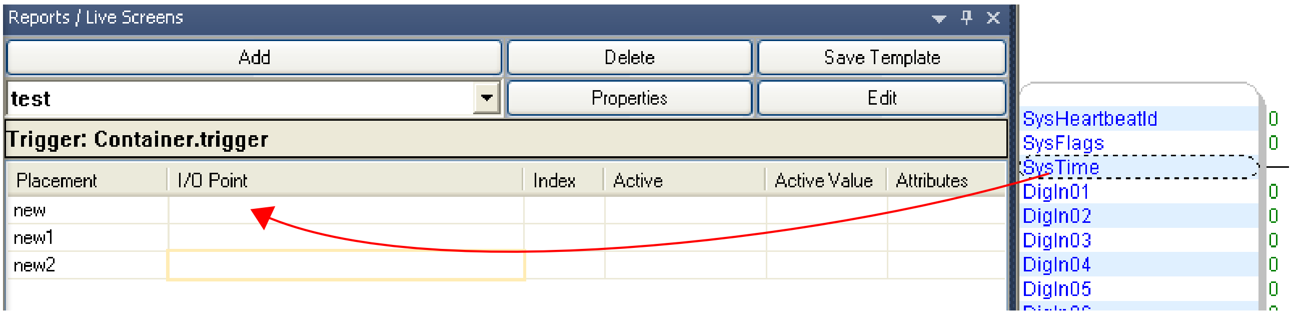


Trigger: <none>

Placement	I/O Point	Index	Active	Active Value	Attributes
new					
new1					
new2					

The user now needs to associate the required application data pin with the relevant placement.

Navigate the application hierarchy, select the container tab that shows the required data pin and simply drag-and-drop that data pin onto the relevant placement.

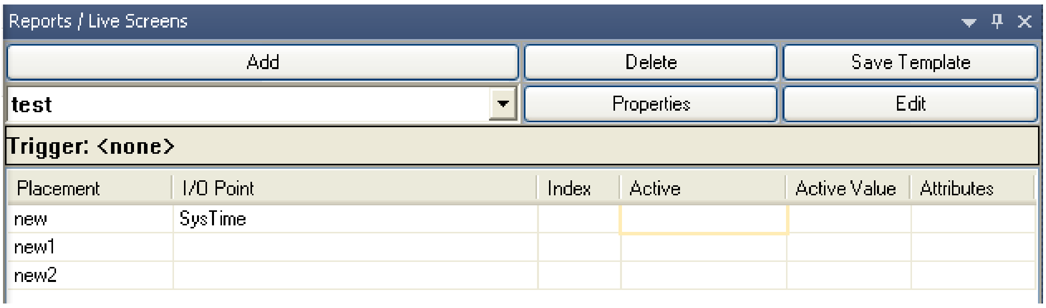


Trigger: Container.trigger

Placement	I/O Point	Index	Active	Active Value	Attributes
new					
new1					
new2					

SysHeartbeatId	0
SysFlags	0
SysTime	
DigIn01	0
DigIn02	0
DigIn03	0
DigIn04	0
DigIn05	0

This will then associate the data pin with the placement, and use that data pin's value on the report whenever the report is triggered.



Trigger: <none>

Placement	I/O Point	Index	Active	Active Value	Attributes
new	SysTime				
new1					
new2					

Dragging an array onto a placement will add the entire array (over-writing the next placements listed until the bottom of the placements list is reached, or all the array elements have been placed).

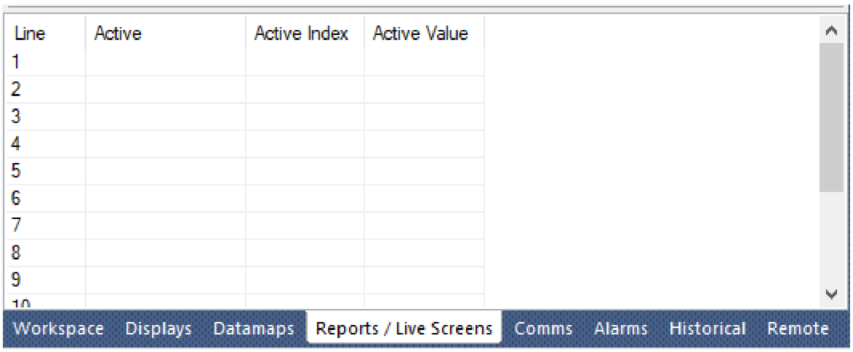
The Object Explorer can also be used to locate and drag pins onto placement items.

Below is a list of the information columns:-

Placement	Shows the placement name.
I/O Point	Shows the path to the application pin assigned to the placement.
Index	For array items, shows the array index to be used.
Active	An active pin can also be set for each placement item, such that the placement item will only be shown on the report / live screen provided the active pin value matches the active value set for that placement. Application pins can be drag-and-dropped onto this column. Dropping an array will
Active Value	Sets the active value to match if this is an “active” placement.

Attributes	This is a general purpose string field that allows the user to assign each placement its own custom “attribute” string. The attribute contents do not affect the application or target device in any way, but allow a host to be informed of any user-defined information that relate to the placement. This attribute string is only transmitted when using XML comms.

Entire report lines can also be assigned active pins, by drag-and-dropping application data pins onto the active line list at the bottom of the Reports / Live Screens window.



Line	Active	Active Index	Active Value
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

When the report / live screen is generated, the report lines will be shown or hidden according to the active pin (plus array index where applicable) and its associated active value.

8.5.1 Report storage

Reports that are stored in the large or small zones are limited to 240 value “slots”, with each slot capable of holding either one double, one integer or up-to eight text bytes.

Any report placement data that exceeds this limit will be ignored and shown as zero when the report is generated.

Note that text strings can also be stored in a report (by dragging a text pin onto the placement I/O point). However, depending on the allocated string size of the I/O point, a string may consume more than one of the 240 possible data values.

A warning will be displayed when a report exceeds the 240 slot storage limit.

This limit does not apply for “live screen” type reports.

8.6 Alarms

Application generated alarms are fully supported by the editor, and any data pin can be configured to raise an alarm. On the pin properties, simply check the Alarm tick box.

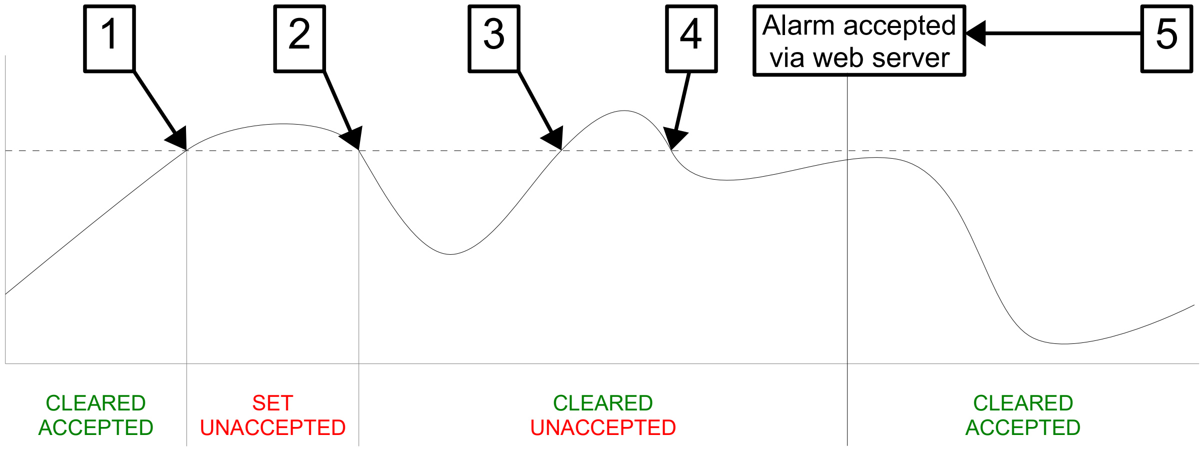
At this point if the pin's value goes from zero to non-zero, an “alarm set” entry will be stored in the alarm log.

When the pin value goes back to zero, an “alarm cleared” entry is stored in the alarm log.

When the alarm itself is accepted (either via Remote Debug mode, the web server or a comms link), an “alarm accepted” entry is stored in the alarm log.

To prevent fleeting alarms from filling up the alarm log, “alarm set” entries are not stored if the alarm has previously been set but not yet accepted.

The diagram below shows the various stages of alarm handling.



The graph shows a temperature input, with the dotted line denoting a “high temperature” trigger value. The alarm status is currently CLEARED and ACCEPTED.

The following stages occur:-

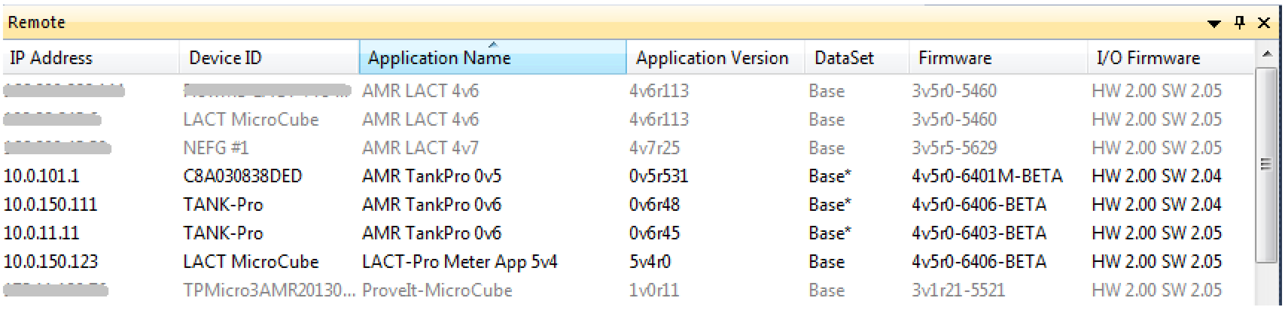
1. Temperature goes above trigger value. Alarm is not currently set or unaccepted, so an “alarm set” entry is stored in the alarm log. Alarm status is SET and UNACCEPTED.
2. Temperature falls below trigger value. Alarm is currently set, so an “alarm cleared” entry is stored in the alarm log. Alarm status is now CLEAR but still UNACCEPTED.
3. Temperature goes above trigger value again. Since the alarm is currently unaccepted, no new “alarm set” entry is logged. Alarm status is still CLEAR but still UNACCEPTED.
4. Temperature goes below trigger value again. Since the alarm is currently unaccepted, no new “alarm cleared” entry is logged. Alarm status is still CLEAR but still UNACCEPTED.
5. Alarm condition is accepted via the web server. An “alarm accepted” entry is stored in the alarm log. Alarm status is now CLEAR and ACCEPTED.

Each entry in the alarm log is timestamped, allowing the user to trace the alarm history.

See also 12.2 - Remote Alarms / Events.

8.7 Remote

This tab provides a list of any local and remote targets that can be accessed for remote debugging or software updates.



Remote

IP Address	Device ID	Application Name	Application Version	DataSet	Firmware	I/O Firmware
[redacted]	[redacted]	AMR LACT 4v6	4v6r113	Base	3v5r0-5460	HW 2.00 SW 2.05
[redacted]	LACT MicroCube	AMR LACT 4v6	4v6r113	Base	3v5r0-5460	HW 2.00 SW 2.05
[redacted]	NEFG #1	AMR LACT 4v7	4v7r25	Base	3v5r5-5629	HW 2.00 SW 2.05
10.0.101.1	C8A030838DED	AMR TankPro 0v5	0v5r531	Base*	4v5r0-6401M-BETA	HW 2.00 SW 2.04
10.0.150.111	TANK-Pro	AMR TankPro 0v6	0v6r48	Base*	4v5r0-6406-BETA	HW 2.00 SW 2.04
10.0.11.11	TANK-Pro	AMR TankPro 0v6	0v6r45	Base*	4v5r0-6403-BETA	HW 2.00 SW 2.05
10.0.150.123	LACT MicroCube	LACT-Pro Meter App 5v4	5v4r0	Base	4v5r0-6406-BETA	HW 2.00 SW 2.05
[redacted]	TPMicro3AMR20130...	ProveIt-MicroCube	1v0r11	Base	3v1r21-5521	HW 2.00 SW 2.05

Each target has the ability to announce its existence on the local network, and all running Visual C|| Cure environments will detect these announcements and automatically add the target to this list.

Remote targets (i.e. those on an external network, such as the Internet, rather than the local network) can also be manually added to the list, but the target details must be manually polled for (using right-click Poll for Details).

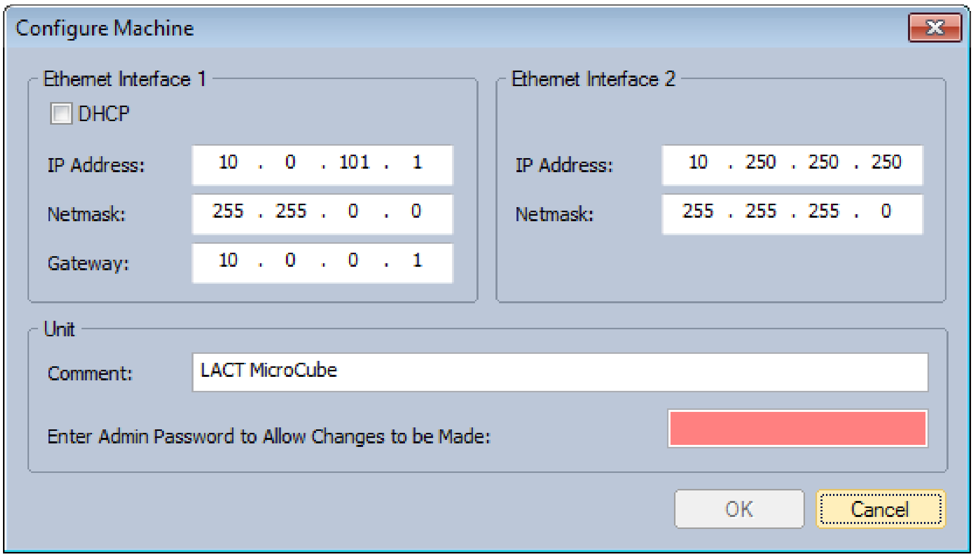
Remote targets will be displayed in grey until “Poll for Details” has been run on that target and comms was established.

Below is a list of the information columns:-

IP Address	Shows the IP address.
Device ID	Shows the hostname.
Application Name	Shows the name of the application installed.
Application Version	Shows the application version.
DataSet	Shows the dataset currently in use. See 6.17 - Datasets for more information.
Firmware	Shows the system firmware version.
I/O Firmware	Shows the I/O firmware version.
System ID	Shows the unique ID.
ETH Port	Shows the network port used to establish comms to the relevant IP address. The target hardware supports two network ports, and, if both are connected and configured correctly, it may be possible to see two entries in the list for a single target (one entry for each port but with different IP addresses).
Comment	Shows the current comment for the target. This can be changed via the right-click Reconfigure Machine dialog (see below).

Various commands can be sent to the targets listed, allowing the user to (e.g.) install new applications, system firmware, I/O firmware or reboot a target.

All these commands can be accessed by right-clicking on a target (or targets) and selecting one of the following:-

Poll for Details	Sends a “get information” message to all selected target(s). A pop-up window will open, showing the list of selected targets. Each target will then be polled and its details updated in the list. If a time-out occurs to any target, the line will be greyed out.
Refresh Local Machines	Sends a “get information” message to all **local** targets.
Remote Debug	Starts a remote debugging session with the target. See 12 - Remote Debug for more information.
Start Web Interface	Opens a web browser and points to the IP address of the select target. This is only available if a single target has been selected.
Configure	Opens the network configuration window to allow the network settings and unit comment of the target to be set.  Configure Machine Ethernet Interface 1 □ DHCP IP Address: 10 . 0 . 101 . 1 Netmask: 255 . 255 . 0 . 0 Gateway: 10 . 0 . 0 . 1 Ethernet Interface 2 IP Address: 10 . 250 . 250 . 250 Netmask: 255 . 255 . 255 . 0 Unit Comment: LACT MicroCube Enter Admin Password to Allow Changes to be Made: Network port 1 can be set to DHCP or Static IP mode. Network port 2 is fixed in Static IP mode. A general purpose “comment” field can also be set. Finally an administrator password for the target must be entered. Click OK to send the new settings to the target. Note : this operation relies on broadcasts, so no feedback is given to confirm that action has been successfully completed. It is recommended that the user double-check that the target device reappears in the remote targets list. Note : RFC950 (http://tools.ietf.org/html/rfc950) states that the IP Address for the two network ports must occupy separate subnets. Care must be taken especially when DHCP is used.
Strobe Ident Lights	This command can be used to identify a target. When sent, this command causes the selected target's red/green user lights to flash orange for 5 seconds.
Add/Remove Machine(s)	Opens a dialog box allowing the user to add or remove remote targets.

Restart	This sub-menu has three options available:- **Warm Start** Restarts the target retaining all current persistent values. **Cold Start Keep Totals** As per Warm Start, but only those persistent pins contained in a group called TOTALS will have their values retained. All other persistent pins will be reset back to their default value (as defined by the application). **Cold Start Clear Totals** Restarts the target resetting all persistent pins back to their default value (as defined by the application). When selected, a pop-up dialog is shown, confirming the IP address of the target. Any administrator password for the target must also be entered. Click OK to restart the target.
Install/Retrieve Files sub-menu	
Install Loaded Application	Installs the currently loaded application to the target. See 10 - Installing Applications for more information.
Get a Backup of the Running Application	Obtains a backup of the application installed in the target. See 13 - Backing Up Applications for more information.
Get Kernel Log	Using this option, the user is able to retrieve the Operating System event log. This is not required in normal operation, and is only required for debugging purposes. This is only available if a single target has been selected.
Update Firmware	Enables the user to install new system firmware to the target. See 14 - Updating Firmware for more information.
Update I/O Firmware	Enables the user to install new I/O firmware to the target. See 14 - Updating Firmware for more information.

8.8 Historical

The target hardware has the ability to store the history of a number of data points on a per minute (or per trigger) basis, or whenever any alarm or event occurs.

The "Events" tab allows 13 items to be stored whenever any alarm or event occurs. These items can then be viewed or downloaded via the web browser.

For historical / trending data, there are four available sets as follows:-

Set	Storage size	Notes
1...3	20000 slots	Stored in flash so are retained on restart
4	172800 slots	This set allows data to be stored on every cycle (i.e. twice per second) for 24 hours, allowing for detailed analysis of application data. However, to enable this, the data is stored in RAM and so will be lost on restart.

To setup the historical data slots, simply drag the required data pins from the application onto any of the free slots.

If an array is dragged, a prompt will ask if the whole array is to be added.

If No is chosen, a single slot will be used, with the array index value defaulting to 0. At this point, the array index can be adjusted to select the required array item using the up / down spinner.

If Yes is chosen, all the array items (up to a maximum of 13 items) will be added starting at the current slot (overwriting any existing data pins), with the index setting automatically incrementing to select the relevant array item.

Alternatively, individual pins can be dragged from the Object Explorer.

Note that only double and integer values are supported, so text pins cannot be stored.

The Trigger setting allows the application itself to determine when to store a new historical slot. Simply drag a pin from the application onto the Trigger cell. Whenever this pin is non-zero, a new historical slot will be stored.

To clear the trigger, click on the Clear button. This resets the storage trigger to the default “per minute” setting.

For Sets 1-3, a Maximum Average Trigger Rate of once every 5 seconds must be respected to guarantee a 25 year life for the target hardware in worst case conditions.

Caution : FLASH memory technology is used to store the historical data. This technology has a limited “lifetime”, so to prevent damaging the FLASH memory, each historical slot should not be written **on average** more than once per day. Hardware faults due to over-use of the FLASH memory will invalidate any warranty. Usage counters are stored in non-volatile memory to allow monitoring over the lifetime of the device.

9 Configuring Targets without Visual C||Cure

For many end-users who just want to install / backup applications, Visual C||Cure is overkill.

To this end, an alternative tool, called NANOconf, is available.

This utility allows uses to install and backup applications without having to open up the application itself (therefore further protecting the IP contained within the application).

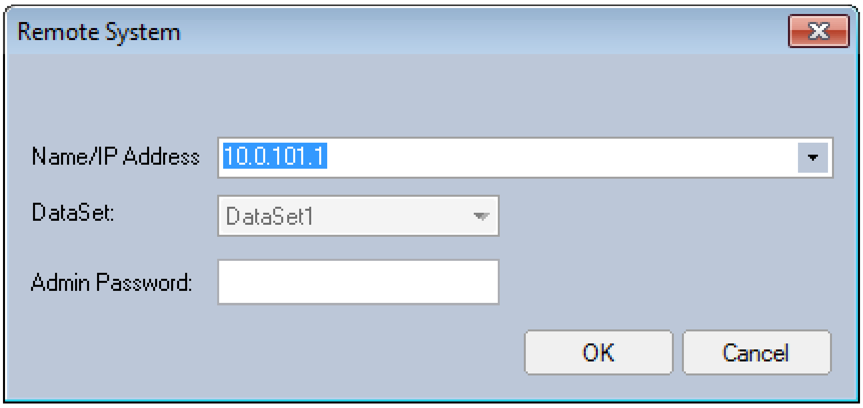
Please contact your supplier for more information.

10 Installing Applications

Once an application is loaded into Visual C||Cure (either by opening an existing application, or writing and saving a new application), it can be installed onto a target.

From the Remote tab, select the required target, right-click and select Install Loaded Application (or select File | Install Application to a Target...).

At this point the Remote System dialog window will be shown:-



Remote System

Name/IP Address 10.0.101.1

DataSet: DataSet1

Admin Password:

Confirm the IP address of the target to update, select the DataSet to be used, and then enter any admin password for the target.

Now click OK, and Visual C||Cure will:-

1. Connect to the target
2. Stop any existing application
3. Transfer the loaded application
4. Restart the target using the new application

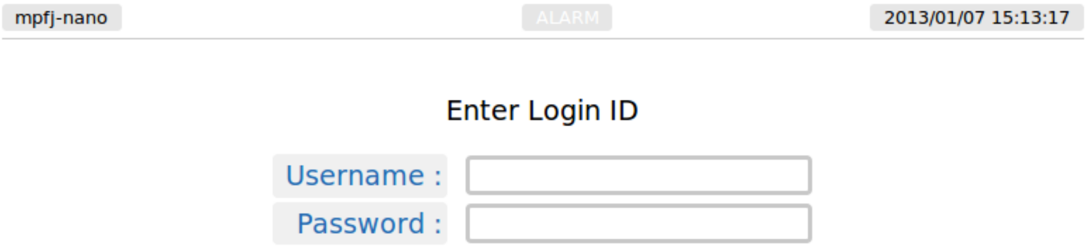
The Remote tab can be used to confirm the progress of the installation, with the App column text updating to show the current progress.

11 Web server

By default, each target contains a built-in web server, which displays an HTML view of the application's configured display tree (see 8.2 - Displays).

11.1 Login Page

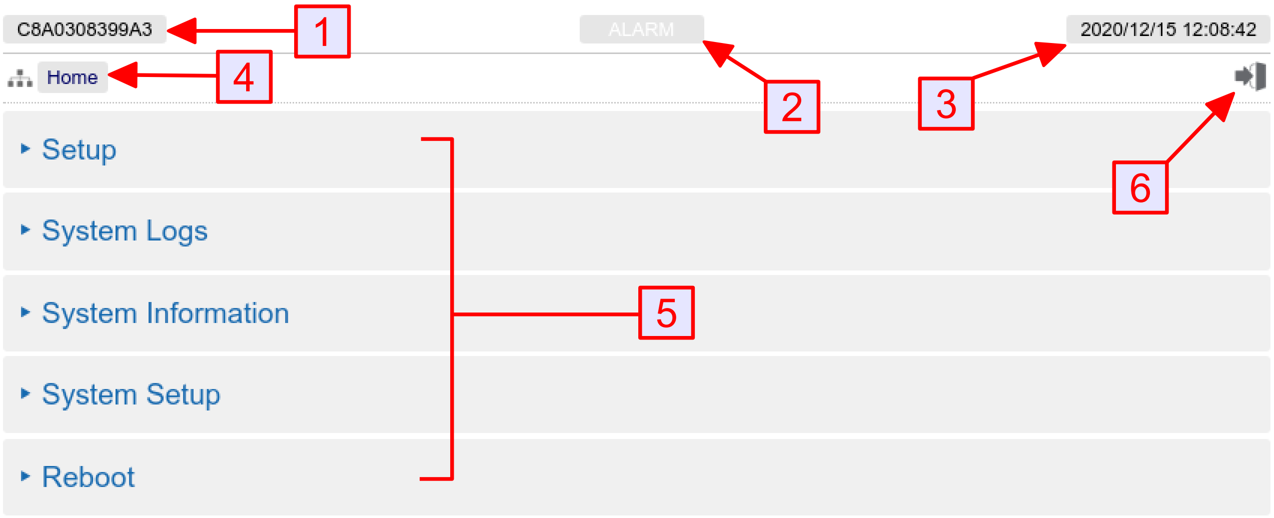
Pointing a browser at the target's IP address will bring up the login screen.



A valid username and password can now be entered. Pressing the Enter key will then send the login details to the target. If login is successful, the Home page will be shown.

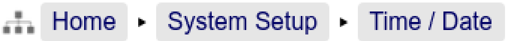
11.2 Home Page

The Home page shows various system items, along with any application display menus and various system menus. Below is an example page, but note that different menu items may be shown depending on how the application has been designed / configured.



The following items are shown:-

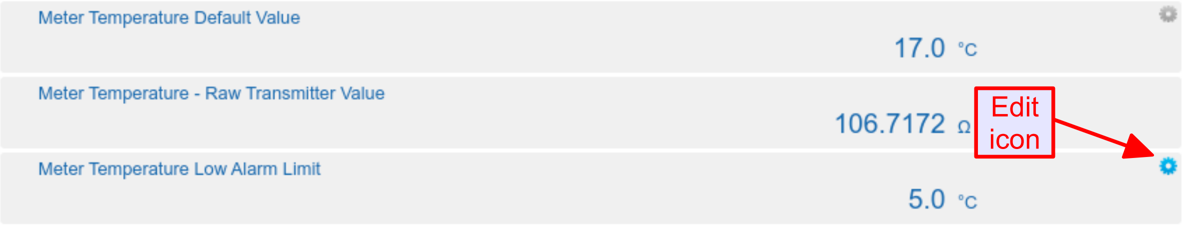
1	Target hostname	This shows the current hostname of the unit. If a hostname has not been set, the system ID will be displayed instead.
2	Alarm indicator	Clicking on this item will jump to the Alarms page. The colour of the alarm indicator shows the current alarm status:- **Flashing red** There are unaccepted alarms. **Solid red** There are only accepted alarms, but not all have cleared. **Grey** There are no alarms present. See 11.3 - Alarms for more information.
3	Time / Date	Clicking on this item will jump to the Time / Date settings page (see 11.5.3 - Time / Date Settings).

4	Navigation breadcrumb trail	This shows the Sitemap icon (see 11.4 - Sitemap) and the hierarchical location of the current page. For example:-  Clicking on any of the breadcrumb items will jump to the relevant page.
5	Application display menu(s)	Any display menus specified in the application will be shown. The application can be configured with a mix of User menus (see 8.2 - Displays) and System menus (see 11.5 - System Menus).
6	Logout	Clicking on this icon will log out the current user.

Clicking on any menu item will navigate into the menu and display the sub-menu items or data point items, as configured in the application's displays tree.

If data point items can be changed (and the user has the security permission to do so), the Edit icon will be displayed.

If the item can be edited, but the user does not have the correct permissions, the Edit icon will be greyed out.

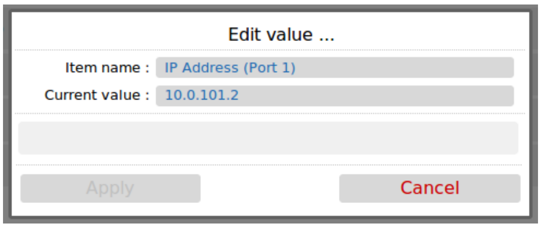


Meter Temperature Default Value 17.0 °C

Meter Temperature - Raw Transmitter Value 106.7172 Ω

Meter Temperature Low Alarm Limit 5.0 °C

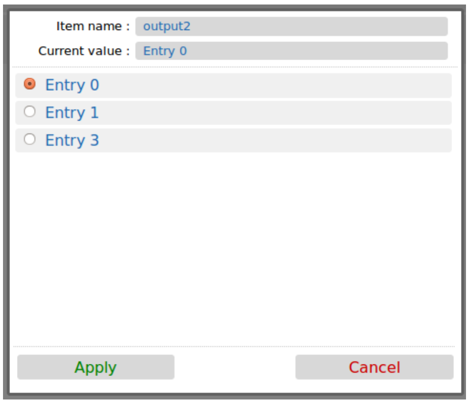
Clicking on this icon will bring up the relevant edit window (number, string or list):-



Edit value ...

Item name : IP Address (Port 1)

Current value : 10.0.101.2



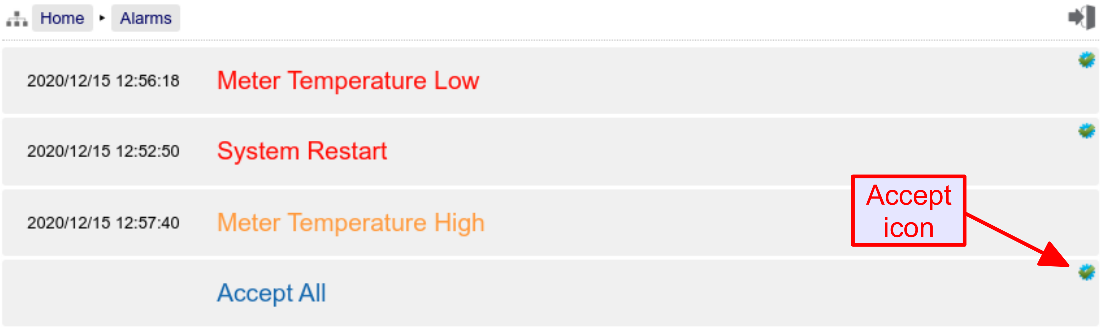
Item name : output2

Current value : Entry 0

● Entry 0
○ Entry 1
○ Entry 3

11.3 Alarms

The Alarms page lists any currently active alarms, along with the time-stamp of when the alarm was set.



2020/12/15 12:56:18 Meter Temperature Low

2020/12/15 12:52:50 System Restart

2020/12/15 12:57:40 Meter Temperature High

Accept All

Alarms shown in red are currently set and have not been accepted.

Alarms shown in orange have been accepted but the alarm has not yet cleared.

See 8.6 - Alarms for more details.

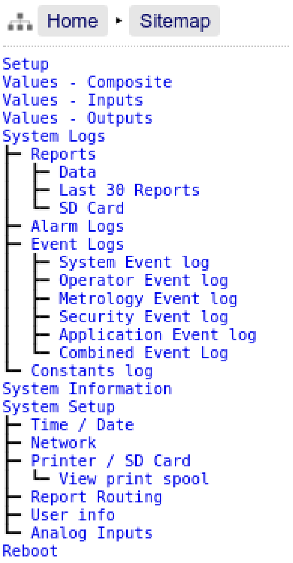
Clicking on the Accept icon will accept the relevant alarm.

Clicking on the icon for “Accept All” will accept all unaccepted alarms.

11.4 Sitemap

Clicking on the Sitemap icon on any webpage will take the user directly to the Sitemap screen.

This allows quick direct access to the web pages configured for the application.



Home ▸ Sitemap

```
Setup
Values - Composite
Values - Inputs
Values - Outputs
System Logs
├ Reports
│ ├ Data
│ ├ Last 30 Reports
│ └ SD Card
├ Alarm Logs
├ Event Logs
│ ├ System Event log
│ ├ Operator Event log
│ ├ Metrology Event log
│ ├ Security Event log
│ ├ Application Event log
│ └ Combined Event Log
└ Constants log
System Information
System Setup
├ Time / Date
├ Network
├ Printer / SD Card
│ └ View print spool
├ Report Routing
├ User info
└ Analog Inputs
Reboot
```

11.5 System Menus

The displays editor in Visual C||Cure allows various system menus (as described below) to be added to the display tree.

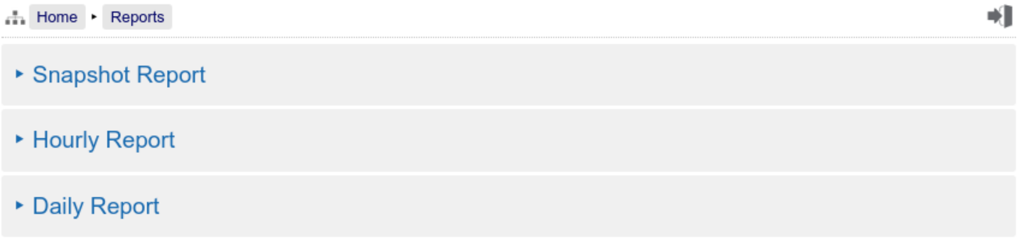
If no system menus are added by the application designer, a default set of system menus will be shown in the following layout:-

```
System Logs
┣ Reports
┣ Alarm Logs
┣ Event Logs
┃ ┣ System Event log
┃ ┣ Operator Event log
┃ ┣ Metrology Event log
┃ ┣ Security Event log
┃ ┣ Application Event log
┃ ┗ Combined Event Log
┗ Constants log
System Information
System Setup
┣ Time / Date
┣ Network
┣ Printer / SD Card
┣ Report Routing
┗ User info
Reboot
```

11.5.1 System Logs

11.5.1.1 Reports

All configured application reports are listed here, for example:-

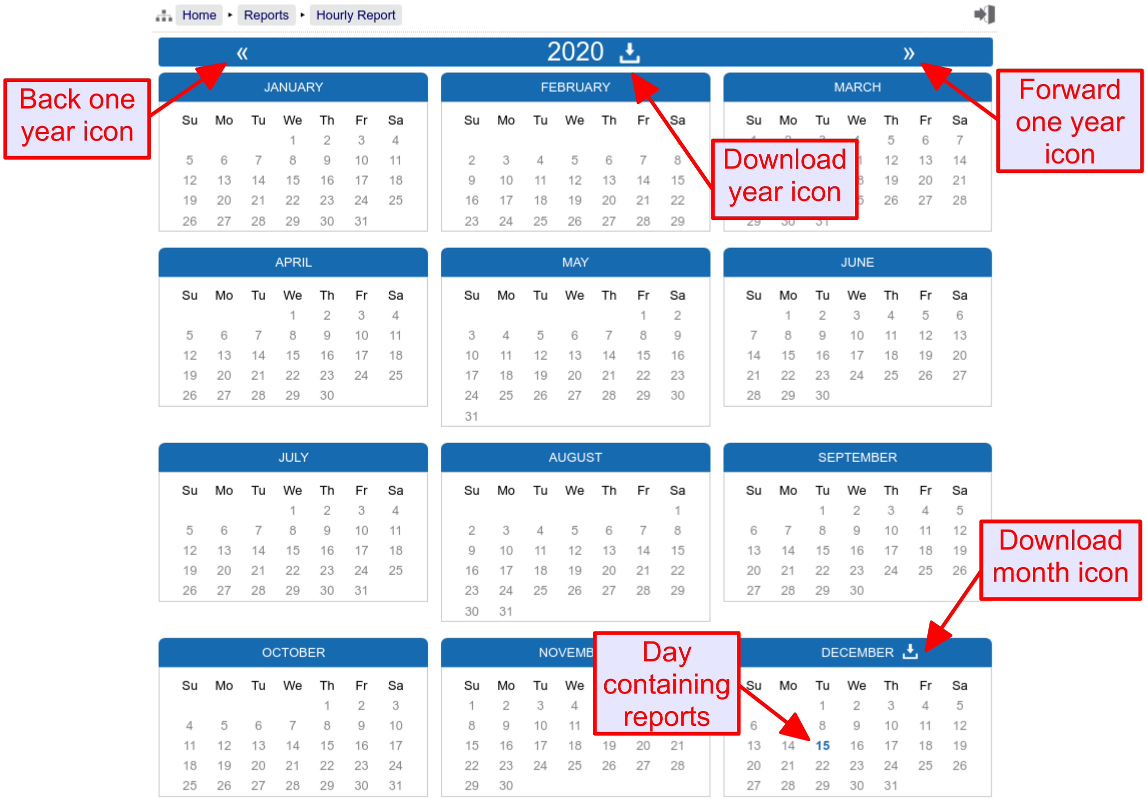


Clicking on a report item will then load up a calendar view.

The year can be changed by clicking on the left and right arrow icons in the header.

The entire set of matching reports for the year can be downloaded in a single ZIP archive by clicking on the download icon shown next to the year.

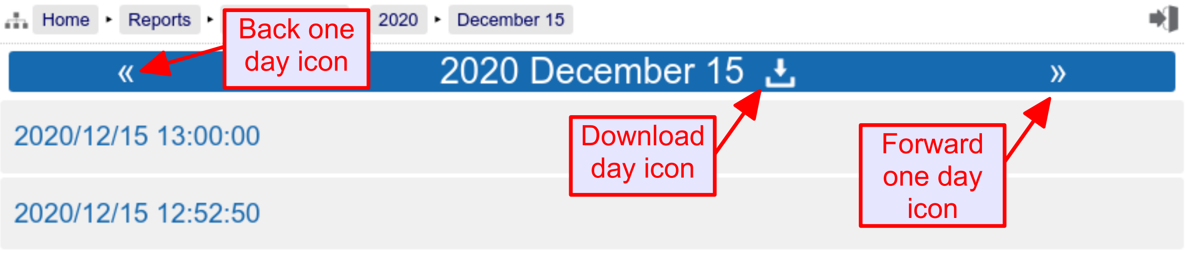
The entire set of matching reports for any particular month can be downloaded in a single ZIP archive by clicking on the download icon shown next to the month name.



Any days containing a matching report will be shown in blue. If a calendar day contains no matching reports, it will be shown in grey.

In the example above, only the 15th December contains an Hourly report.

Clicking on a day with reports will open a new page listing the timestamps for all matching reports.



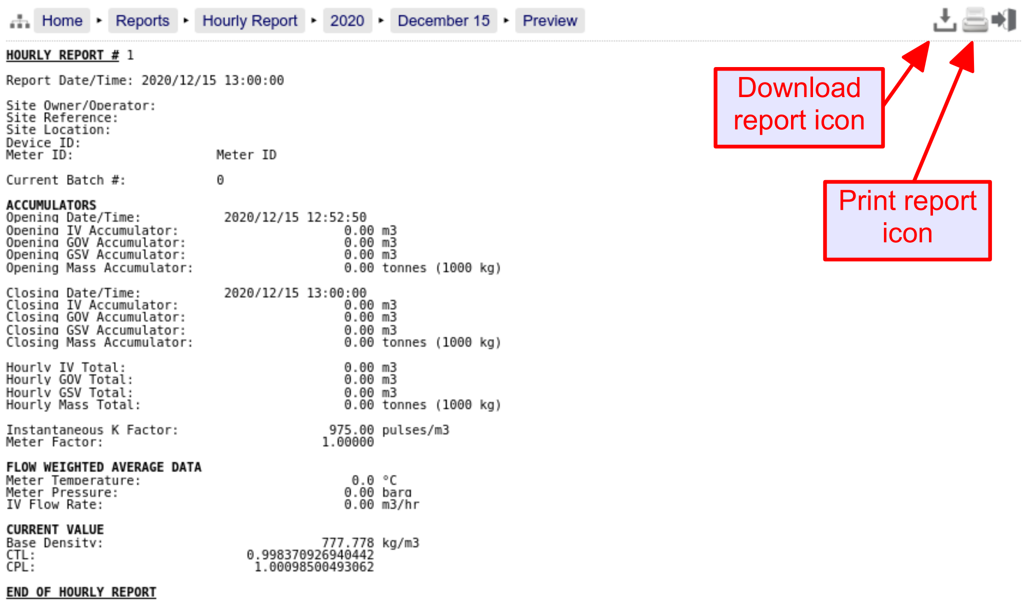
2020/12/15 13:00:00

2020/12/15 12:52:50

The day can be changed by clicking on the left and right arrow icons in the header.

The entire set of matching reports for the day can be downloaded in a single ZIP archive by clicking on the download icon shown next to the year.

Clicking on a particular report item will display a copy of the report itself.

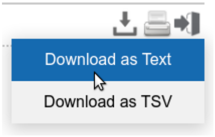


```
HOURLY REPORT # 1

Report Date/Time: 2020/12/15 13:00:00

Site Owner/Operator:
Site Reference:
Site Location:
Device ID:
Meter ID:                   Meter ID

Current Batch #:            0

ACCUMULATORS
Opening Date/Time:           2020/12/15 12:52:50
Opening IV Accumulator:                     0.00 m3
Opening GOV Accumulator:                    0.00 m3
Opening GSV Accumulator:                    0.00 m3
Opening Mass Accumulator:                   0.00 tonnes (1000 kg)

Closing Date/Time:           2020/12/15 13:00:00
Closing IV Accumulator:                     0.00 m3
Closing GOV Accumulator:                    0.00 m3
Closing GSV Accumulator:                    0.00 m3
Closing Mass Accumulator:                   0.00 tonnes (1000 kg)

Hourly IV Total:                            0.00 m3
Hourly GOV Total:                           0.00 m3
Hourly GSV Total:                           0.00 m3
Hourly Mass Total:                          0.00 tonnes (1000 kg)

Instantaneous K Factor:                   975.00 pulses/m3
Meter Factor:                            1.00000

FLOW WEIGHTED AVERAGE DATA
Meter Temperature:                           0.0 °C
Meter Pressure:                             0.00 barg
IV Flow Rate:                               0.00 m3/hr

CURRENT VALUE
Base Density:                            777.778 kg/m3
CTL:                           0.998370926940442
CPL:                            1.00098500493062

END OF HOURLY REPORT
```

The report can be printed to the Web Printer by clicking on the Print icon in the header.

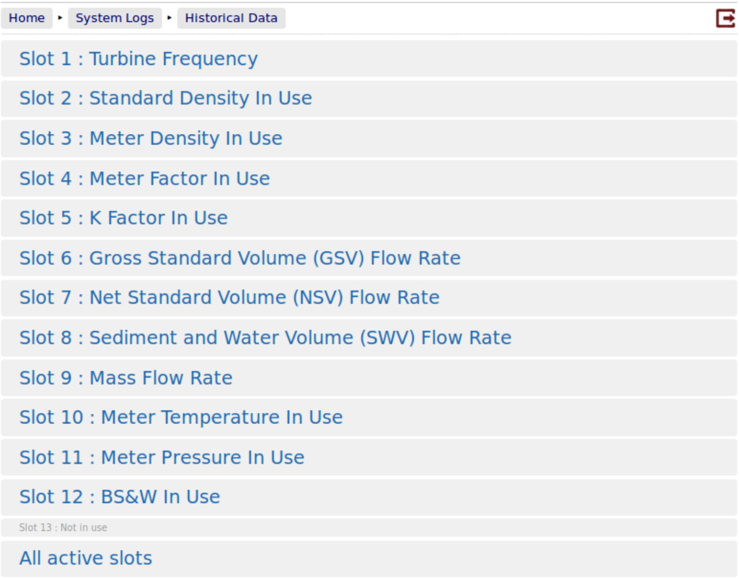
The report can be downloaded by clicking the Download icon in the header.

If configured in the application, various report download options may be shown in a drop-down box. If no drop-down is shown, a plain text file will be downloaded.



11.5.1.2 Historical Data

The Historical Data page shows a list of all the application's Historical Data pins (see 8.8 - Historical).



Slot 1 : Turbine Frequency

Slot 2 : Standard Density In Use

Slot 3 : Meter Density In Use

Slot 4 : Meter Factor In Use

Slot 5 : K Factor In Use

Slot 6 : Gross Standard Volume (GSV) Flow Rate

Slot 7 : Net Standard Volume (NSV) Flow Rate

Slot 8 : Sediment and Water Volume (SWV) Flow Rate

Slot 9 : Mass Flow Rate

Slot 10 : Meter Temperature In Use

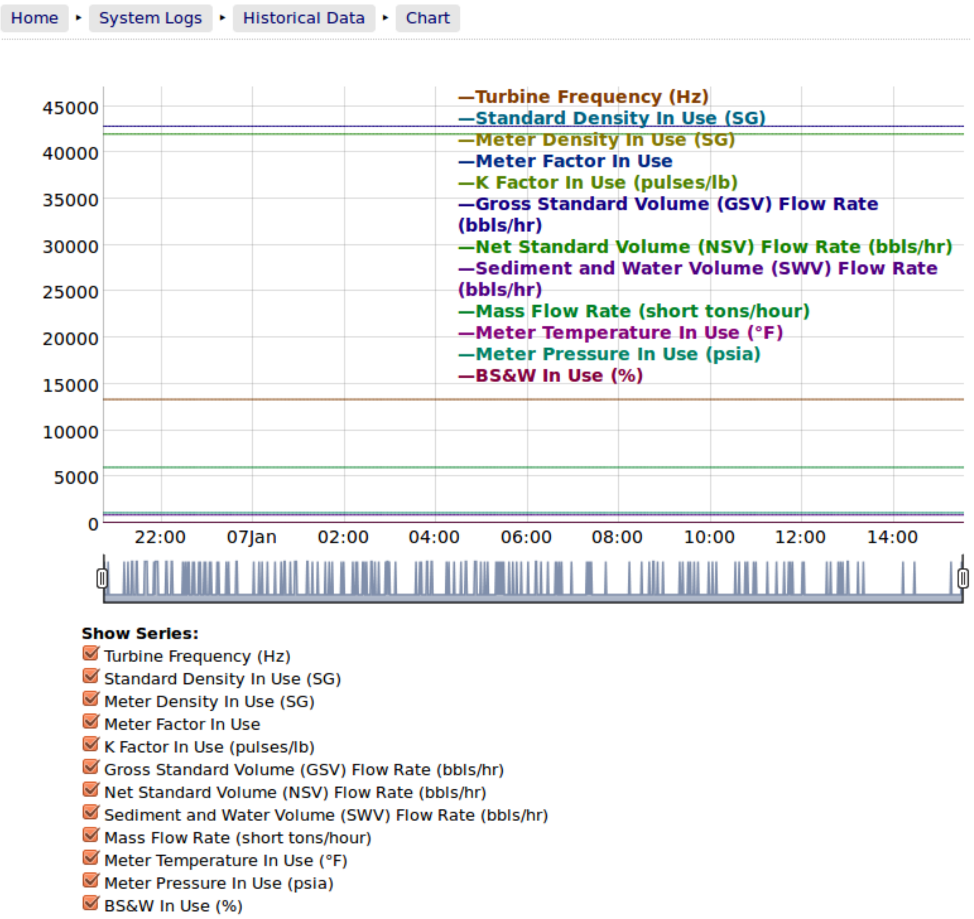
Slot 11 : Meter Pressure In Use

Slot 12 : BS&W In Use

Slot 13 : Not in use

All active slots

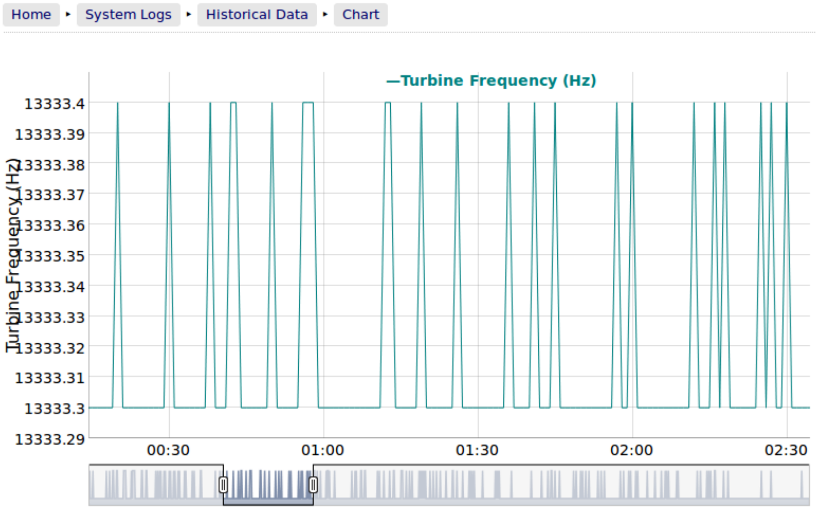
Clicking on any of the pin slots will show a chart of the data stored for that slot.


Home ▸ System Logs ▸ Historical Data ▸ Chart
—Turbine Frequency (Hz)
—Standard Density In Use (SG)
—Meter Density In Use (SG)
—Meter Factor In Use
—K Factor In Use (pulses/lb)
—Gross Standard Volume (GSV) Flow Rate (bbls/hr)
—Net Standard Volume (NSV) Flow Rate (bbls/hr)
—Sediment and Water Volume (SWV) Flow Rate (bbls/hr)
—Mass Flow Rate (short tons/hour)
—Meter Temperature In Use (°F)
—Meter Pressure In Use (psia)
—BS&W In Use (%)
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
22:00
07Jan
02:00
04:00
06:00
08:00
10:00
12:00
14:00

Show Series:

☑ Turbine Frequency (Hz)
☑ Standard Density In Use (SG)
☑ Meter Density In Use (SG)
☑ Meter Factor In Use
☑ K Factor In Use (pulses/lb)
☑ Gross Standard Volume (GSV) Flow Rate (bbls/hr)
☑ Net Standard Volume (NSV) Flow Rate (bbls/hr)
☑ Sediment and Water Volume (SWV) Flow Rate (bbls/hr)
☑ Mass Flow Rate (short tons/hour)
☑ Meter Temperature In Use (°F)
☑ Meter Pressure In Use (psia)
☑ BS&W In Use (%)

By default, the entire available time range is displayed. However this can be adjusted using the slider handles at the bottom of the chart. This allows the user to narrow or expand the time range displayed, and slide the time window left / right.

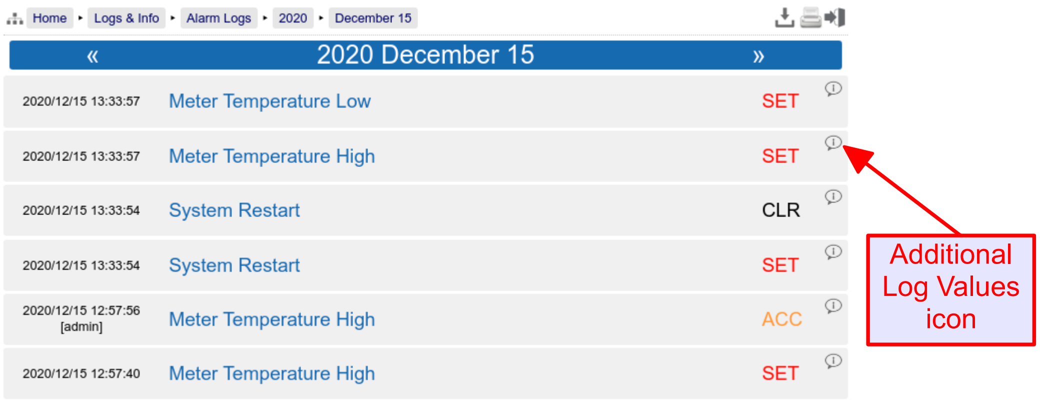


The “All Active Slots” item displays a chart showing all the data slots.

The check boxes at the foot of the chart allow individual data series to be show / hidden.

11.5.1.3 Alarm Logs

Similar to the Reports, the Alarm Logs page shows a calendar view of all alarm log items stored on the target. Clicking on a valid date will show a list of alarm events for that day.



Home ▸ Logs & Info ▸ Alarm Logs ▸ 2020 ▸ December 15

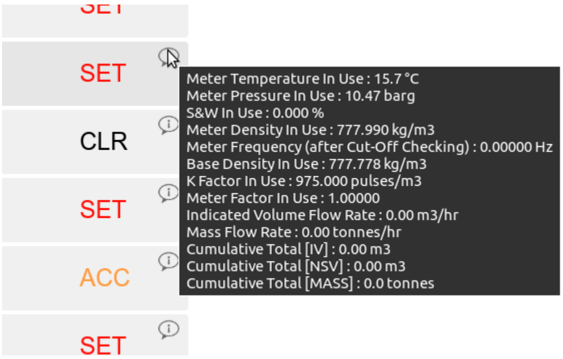
« 2020 December 15 »

2020/12/15 13:33:57	Meter Temperature Low	SET
2020/12/15 13:33:57	Meter Temperature High	SET
2020/12/15 13:33:54	System Restart	CLR
2020/12/15 13:33:54	System Restart	SET
2020/12/15 12:57:56 [admin]	Meter Temperature High	ACC
2020/12/15 12:57:40	Meter Temperature High	SET

The target is able to store a maximum of 1000 alarm log entries, at which point new alarm entries overwrite the oldest entries.

The list can be printed by clicking on the Printer icon or downloaded using the Download icon.

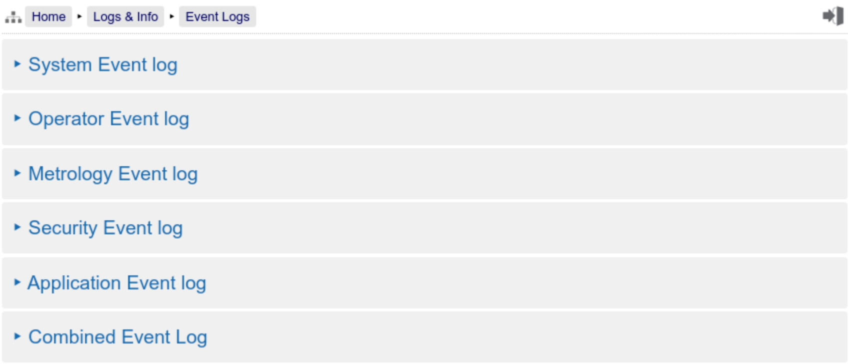
If configured in the application (refer to the "Events" tab in 8.8 - Historical), each entry may also show an "Additional Log Values" (or ALVs) icon. If you hover over this icon, a pop-up box will be displayed showing a set of up-to 13 data points snapshot at the time of the alarm.



Meter Temperature In Use : 15.7 °C
Meter Pressure In Use : 10.47 barg
S&W In Use : 0.000 %
Meter Density In Use : 777.990 kg/m3
Meter Frequency (after Cut-Off Checking) : 0.00000 Hz
Base Density In Use : 777.778 kg/m3
K Factor In Use : 975.000 pulses/m3
Meter Factor In Use : 1.00000
Indicated Volume Flow Rate : 0.00 m3/hr
Mass Flow Rate : 0.00 tonnes/hr
Cumulative Total [IV] : 0.00 m3
Cumulative Total [NSV] : 0.00 m3
Cumulative Total [MASS] : 0.0 tonnes

11.5.1.4 Event Logs

The Event Logs page shows a list of event log types.



Home ▸ Logs & Info ▸ Event Logs

- System Event log
- Operator Event log
- Metrology Event log
- Security Event log
- Application Event log
- Combined Event Log

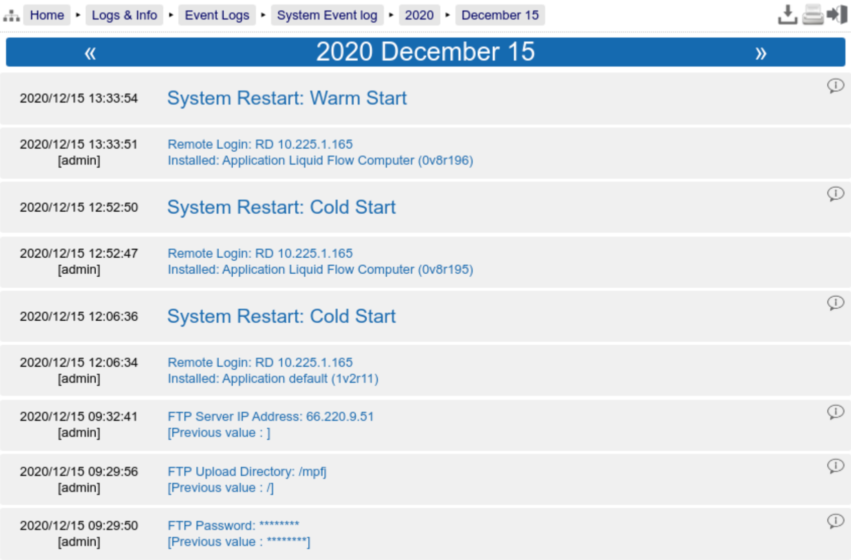
Type	Description	Maximum Entries
System	Any system event such as changing date/time, installing new application or firmware, restarts, network changes.	1000
Operator	Changes to the application values made by end-users during normal operation of the unit.	2500
Metrology	Changes to any application values either:- a) contained in the "Metrology" group (see 6.14 - Tables) b) have "Metrology" write access defined in the Displays tree	1000
Security	Login/logout events via the website or XML comms link.	1000

Application	Information logged via the nfevent() function (see 7.6.4 - Debug Functions).	1000
Combined	This shows a list of the last 30 events of any type.	-

Apart from the Combined Event Log, clicking on a particular event type shows a calendar view (similar to the Alarm Logs page) of all relevant event log items stored on the target. Clicking on a valid date will show the list of events for that day.

The screenshot below shows an example of the System Event log.

Note that ALVs also supported in all event logs.



Home ▸ Logs & Info ▸ Event Logs ▸ System Event log ▸ 2020 ▸ December 15

« 2020 December 15 »

```
2020/12/15 13:33:54   System Restart: Warm Start

2020/12/15 13:33:51   Remote Login: RD 10.225.1.165
     [admin]          Installed: Application Liquid Flow Computer (0v8r196)

2020/12/15 12:52:50   System Restart: Cold Start

2020/12/15 12:52:47   Remote Login: RD 10.225.1.165
     [admin]          Installed: Application Liquid Flow Computer (0v8r195)

2020/12/15 12:06:36   System Restart: Cold Start

2020/12/15 12:06:34   Remote Login: RD 10.225.1.165
     [admin]          Installed: Application default (1v2r11)

2020/12/15 09:32:41   FTP Server IP Address: 66.220.9.51
     [admin]          [Previous value : ]

2020/12/15 09:29:56   FTP Upload Directory: /mpfj
     [admin]          [Previous value : /]

2020/12/15 09:29:50   FTP Password: ********
     [admin]          [Previous value : ********]
```

The target is able to store a maximum number of event log entries, as per the table above. When the log is full, new events overwrite the oldest entries of that event type.

The list can be printed by clicking on the Printer icon or downloaded using the Download icon.

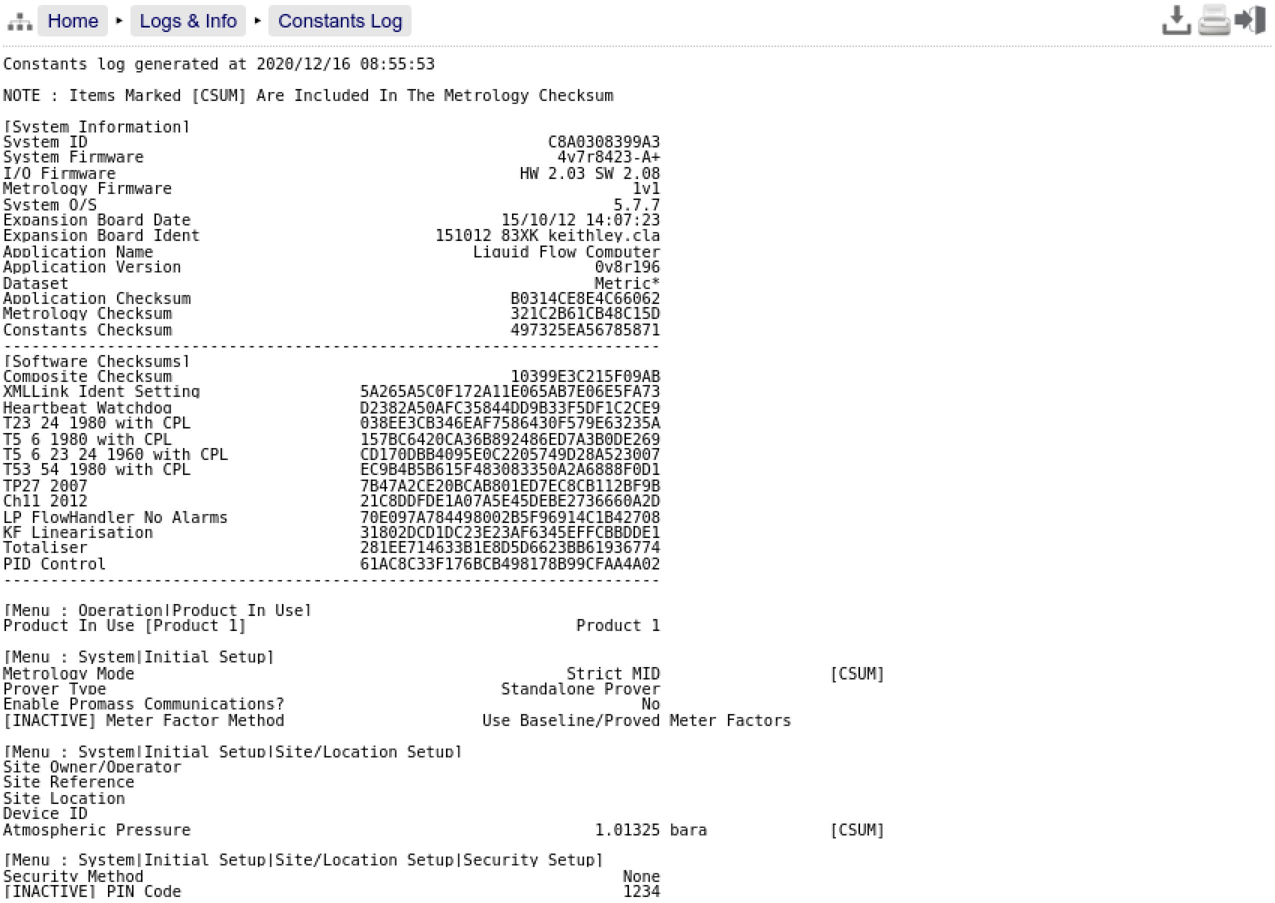
11.5.1.5 Constants Log

The Constants Log page shows a list of the current values of all persistent pins contained in the application display tree.

Also shown is list of System Information and Software Checksums as defined in 11.5.2 - System Information.

Items marked as [INACTIVE] at the beginning of the line are currently hidden from the application menus but are still included for reference purposes.

The [CSUM] at the end of the line indicates that item can only be changed when logged in with Metrology level access and the NMI/MET hardware link is active.



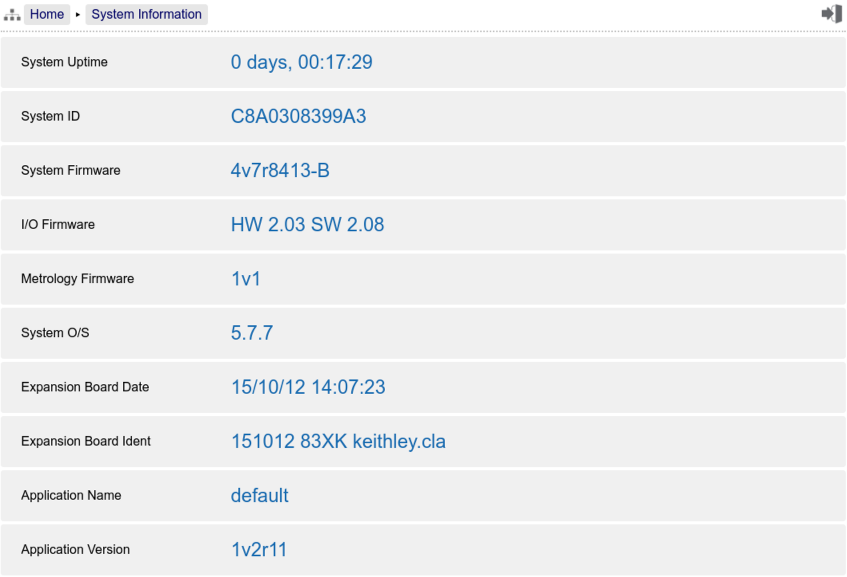
```
Constants log generated at 2020/12/16 08:55:53

NOTE : Items Marked [CSUM] Are Included In The Metrology Checksum

[System Information]
System ID                                                 C8A0308399A3
System Firmware                                            4v7r8423-A+
I/O Firmware                                           HW 2.03 SW 2.08
Metrology Firmware                                                 1v1
System O/S                                                       5.7.7
Expansion Board Date                                 15/10/12 14:07:23
Expansion Board Ident                         151012 83XK keithley.cla
Application Name                                  Liquid Flow Computer
Application Version                                            0v8r196
Dataset                                                        Metric*
Application Checksum                                  B0314CE8E4C66062
Metrology Checksum                                    321C2B61CB48C15D
Constants Checksum                                    497325EA56785871
----------------------------------------------------------------------
[Software Checksums]
Composite Checksum                                    10399E3C215F09AB
XMLLink Ident Setting                 5A265A5C0F172A11E065AB7E06E5FA73
Heartbeat Watchdog                    D2382A50AFC35844DD9B33F5DF1C2CE9
T23 24 1980 with CPL                  038EE3CB346EAF7586430F579E63235A
T5 6 1980 with CPL                    157BC6420CA36B892486ED7A3B0DE269
T5 6 23 24 1960 with CPL              CD170DBB4095E0C2205749D28A523007
T53 54 1980 with CPL                  EC9B4B5B615F483083350A2A6888F0D1
TP27 2007                             7B47A2CE20BCAB801ED7EC8CB112BF9B
Ch11 2012                             21C8DDFDE1A07A5E45DEBE2736660A2D
LP FlowHandler No Alarms              70E097A784498002B5F96914C1B42708
KF Linearisation                      31802DCD1DC23E23AF6345EFFCBBDDE1
Totaliser                             281EE714633B1E8D5D6623BB61936774
PID Control                           61AC8C33F176BCB498178B99CFAA4A02
----------------------------------------------------------------------

[Menu : Operation|Product In Use]
Product In Use [Product 1]                                   Product 1

[Menu : System|Initial Setup]
Metrology Mode                                              Strict MID                 [CSUM]
Prover Type                                          Standalone Prover
Enable Promass Communications?                                      No
[INACTIVE] Meter Factor Method                     Use Baseline/Proved Meter Factors

[Menu : System|Initial Setup|Site/Location Setup]
Site Owner/Operator
Site Reference
Site Location
Device ID
Atmospheric Pressure                                           1.01325 bara            [CSUM]

[Menu : System|Initial Setup|Site/Location Setup|Security Setup]
Security Method                                                   None
[INACTIVE] PIN Code                                               1234
```

The list can be printed by clicking on the Printer icon or downloaded using the Download icon.

11.5.2 System Information

The System Information page shows a list of various system information items.

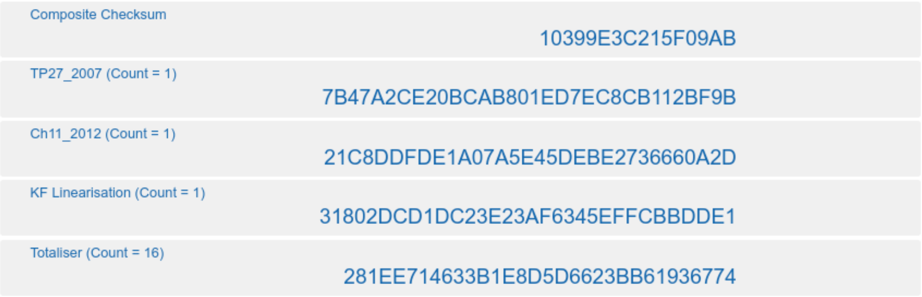
Below a screenshot of the first few lines:-

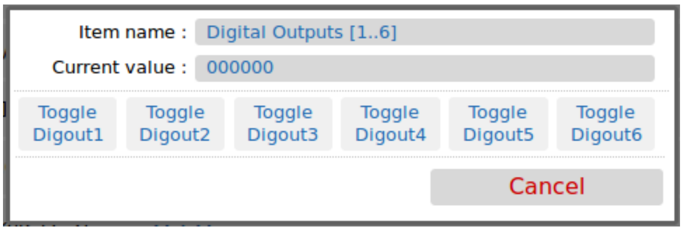


System Uptime	0 days, 00:17:29
System ID	C8A0308399A3
System Firmware	4v7r8413-B
I/O Firmware	HW 2.03 SW 2.08
Metrology Firmware	1v1
System O/S	5.7.7
Expansion Board Date	15/10/12 14:07:23
Expansion Board Ident	151012 83XK keithley.cla
Application Name	default
Application Version	1v2r11

The following items are shown:-

System Uptime	Elapsed time since the unit was last powered on or rebooted.
System ID	The unique ID of the target.
System Firmware	The system firmware version.

I/O Firmware	If applicable, the I/O firmware version of the target.
Metrology Firmware	The Metrology firmware version.
System O/S	The Operating System version.
Expansion Board Date	If applicable, the calibration date of the expansion board
Expansion Board Ident	If applicable, the identification string of the expansion board
Application Name	The name of the application.
Application Version	The application version.
Dataset	The name of the dataset current in use. If any of the dataset values have been changed since the application was installed, an asterisk will be shown at the end of the name.
Application Checksum	The application checksum. When Visual C||Cure saves an application, it is possible to set the Application Version number manually. However, when the application is saved, a checksum value is calculated that is unique to the layout, connectivity and content of the application. Any changes to the application will cause the checksum to change. This provides an additional verification method, for example, when performing Factory Acceptance Tests.
Metrology Checksum	The checksum of the current values of the application's persistent pins that are either:- (a) Contained in the Metrology group (b) Have been added to the “Remote” displays tree and have “Metrology” security write access
Constants Checksum	The checksum of the current values of the application's persistent pins that are changeable via the “Remote” displays tree.
Software Checksums	If applicable, this sub-menu lists the checksums of any Metrology blocks. The first entry is a composite value that combines all the other checksums into a single value to allow the user to quickly check if any block has changed. For example:-  Composite Checksum 10399E3C215F09AB TP27_2007 (Count = 1) 7B47A2CE20BCAB801ED7EC8CB112BF9B Ch11_2012 (Count = 1) 21C8DDFDE1A07A5E45DEBE2736660A2D KF Linearisation (Count = 1) 31802DCD1DC23E23AF6345EFFCBBDDE1 Totaliser (Count = 16) 281EE714633B1E8D5D6623BB61936774

Profile	This item shows the application cycle time (in milliseconds). Four values are shown:- Min the minimum cycle time since power-on. Last the cycle time for the last cycle completed. Avg the average cycle time since power-on. Max the maximum cycle time since power-on. Clicking on the Edit icon allows the user to reset the profile information.
Digital Inputs	If Digital Inputs are configured in the application, this item shows the current values.
Digital Outputs	If Digital Outputs are configured in the application, this item shows the current settings. The outputs can be temporarily over-written by clicking on the Edit icon, which brings up the following dialog:-  Item name : Digital Outputs [1..6] Current value : 000000 Toggle Digout1 Toggle Digout2 Toggle Digout3 Toggle Digout4 Toggle Digout5 Toggle Digout6 Cancel Clicking on the toggle items will change the Digital Output setting (over-writing the application's value). Click Cancel to close the dialog and return control of the Digital Outputs back to the application.
Pulse Outputs Buckets	If Pulse Outputs are configured in the application, this item shows the number of pulses still to be output from the hardware.

11.5.3 Time / Date Settings

11.5.3.1 Time / Date

The following system settings are shown:-

TimeZone	If configured in the application, this line will show the currently selected Time Zone.
Time Offset (HH:MM)	Shows the current time offset (in hours:minutes).
Date Format	Shows the current date format. The following are available:- **YYYY/MM/DD** e.g. 2012/11/26 **DD/MM/YYYY** e.g. 26/11/2012 **MM/DD/YYYY** e.g. 11/26/2012
Date	Shows the current date (using the format defined below).
Time	Shows the current time.

All settings can be changed by administrator users by clicking on the relevant Edit icon.

11.5.3.2 Daylight Saving Time

Shows whether the Daylight Saving function is enabled or disabled on the target.

If the start date or end date are invalid or the same, the function will be disabled.

Start Date	Shows the start date to enter daylight savings.
End Date	Shows the end date to exit daylight savings.
Changeover Hour	Shows the hour at which to enter/exit daylight savings.

All settings can be changed by administrator users by clicking on the relevant Edit icon.

11.5.3.3 NTP Service

When enabled, the unit time can be synchronised against any NTP server.

The automatic check occurs at 03:33 every day, but the exact second within the minute is not fixed and varies from machine to machine.

The time is not updated if the time difference is less than ½ second or greater than 15 minutes.

Mode	Shows whether the NTP (Network Time Protocol) service is enabled or disabled on the target. The following options are available:- **Disabled** NTP service is disabled. **Automatic Sync** NTP is checked at 03:33 every day. **Sync On Startup** NTP is checked only when the unit is powered on or restarts. **Sync On Startup & Automatic** NTP is checked when the unit is powered on or restarts and at 03:33 every day.
Server	Specifies the IP address of the NTP server to use.
Last Sync	Shows when the NTP service last performed a successful sync, along with the time adjustment applied.
Manual Sync	Allows the user to perform a manual sync. Note that, using this option, the time will always be updated regardless of the time difference.

All settings can be changed by administrator users by clicking on the relevant Edit icon.

11.5.4 Network

11.5.4.1 Ethernet Settings

The Network page shows the current network settings for both network ports.

The settings can be changed by administrator users by clicking on the relevant Edit icon.

If DHCP is chosen as the IP Method, the current in-use settings values are shown, but greyed out as they are for information only.

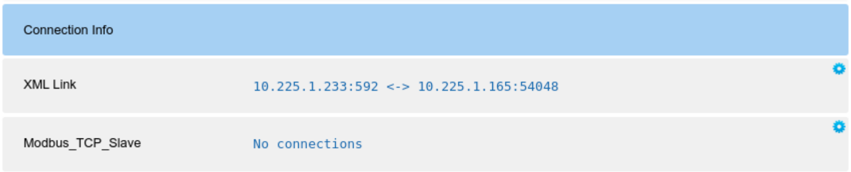
Changed values are shown in red and are only enacted after the “Apply” item has been clicked.

In order to correctly apply changes to the networking, all changes must be made before clicking the “Apply” item.

Once “Apply” has been clicked, the network will re-configure itself at once.

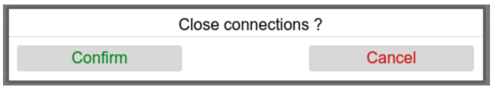
11.5.4.2 Connection Info

If there are any Ethernet comms links configured in the application, the Connection Info section will be visible. This shows any active connection for each comms link. For example:-



Connection Info	
XML Link	10.225.1.233:592 <-> 10.225.1.165:54048
Modbus_TCP_Slave	No connections

Each link has an Edit icon. Clicking on this icon will bring up a dialog box that allows the user to forcibly close all connections for that link.



Close connections ?

11.5.5 Printers / SD Card / FTP

11.5.5.1 Printers

The target hardware supports either serial printers or Ethernet Postscript printers.

Serial printers must be connected to Serial Port 1.

The Baud Rate and Flow Control settings can be configured by any administrator users.

Network printers must have a valid IP address and network port configured. A scaling value can also be applied.

The “Print Test Page” item can be used to test the network printer settings have been entered correctly. This will send a rectangular box of asterisks (80 columns wide by 69 rows high) to the printer, and the scaling value can be used to ensure all 4 sides are visible on the printout.

The “Print Spool” item shows the number of printer jobs waiting to be printed. Clicking on the Edit icon will show a list of all pending printer jobs. Administrator users can delete printer jobs if required.

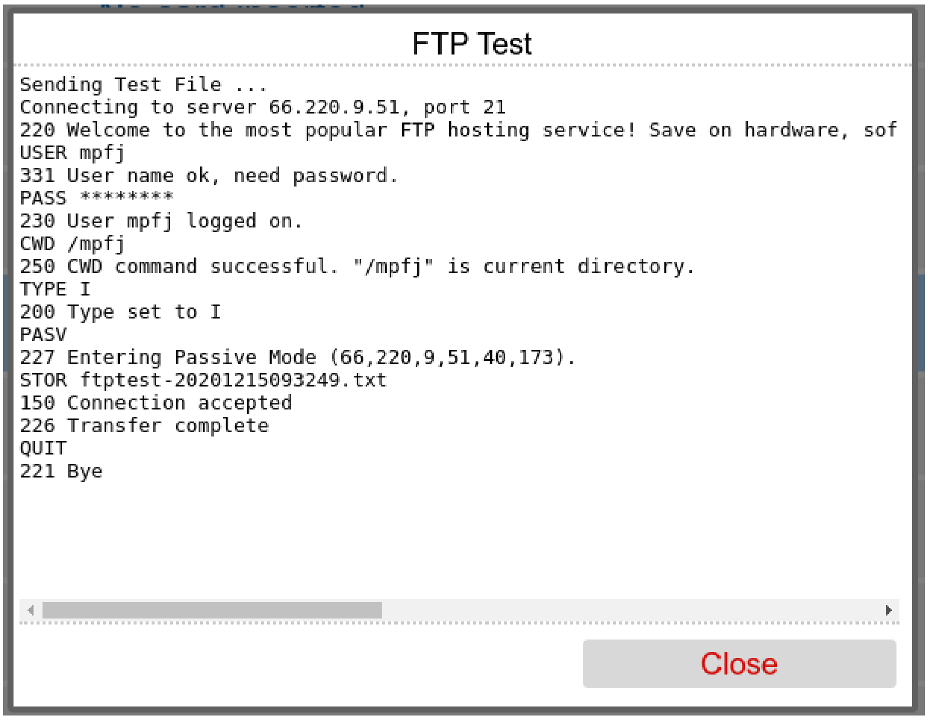
11.5.5.2 SD Card

Status	Shows the status of the SD Card. To install / remove an SD Card, click on the Edit icon and select the relevant option.
Archive Alarms/Events	Sets how often the alarms and events logs are stored to the SD Card. The logs are stored in a ZIP archive in a directory structure of:- <Year> / <Month> / <Date> / <Filename> The options are Daily, Weekly, Monthly or Disabled.
Archive Historicals	Sets how often the historical trending data is stored to the SD Card. The data is stored in a ZIP archive in a directory structure of:- <Year> / <Month> / <Date> / <Filename> The options are Daily, Weekly, Monthly or Disabled.

11.5.5.3 FTP

Server IP Address	Specifies the IP address of the FTP server to use.
Port	Specifies the network port to use.

<table>
<tr><td>Username</td><td>Specifies the login username.</td></tr>
<tr><td>Password</td><td>Specifies the login password.</td></tr>
<tr><td>Upload Directory</td><td>Specify the directory path format when data is uploaded via FTP.
The FTP filename is defined using the “file” report attribute (see 6.6.5.7 - Reports).
Multiple fields can be joined to create a custom path based on (for example) report name, time, date, device ID.
Fields can be embedded within the FTP path, and will be expanded accordingly.
The following fields are available:-
%d Day (01 to 31)
%D Date in current format (%Y%m%d / %d%m%Y / %m%d%Y)
%e Seconds since epoch (decimal)
%E Seconds since epoch (hexadecimal)
%H Hour (00 to 23)
%I Device ID
%m Month (01 to 12)
%M Minute (00 to 59)
%r Report name
%s Device Serial Number
%S Second (00 to 59)
%T Time (%H%M%S)
%y Year without century (00 to 99)
%Y Year with century
%+ Date and Time (%D%T)
%% Literal '%' character
The default format is "%r-%+".
<b>Examples</b>
Text report named "Daily" created on 1st February 2018 at 13:30:00.
Device serial number is C8A0308399A3.
<table>
<tr><th>File path format</th><th>Generated filename</th></tr>
<tr><td>/Site123/%Y</td><td>/Site123/2018/Daily-20180201133000.txt</td></tr>
<tr><td>/Site123/%Y/%r</td><td>/Site123/2018/Daily/Daily-20180201133000.txt</td></tr>
<tr><td>/%s/Reports</td><td>/C8A0308399A3/Reports/Daily-20180201133000.txt</td></tr>
</table></td></tr>
<tr><td>Upload Format</td><td>Sets the upload format when sending reports.
<b>Plain Text</b> Send report in plain text.
<b>TSV</b> Send report in Tab Separated Values format.
<b>TFX/Plain Text</b> If the report supports it, send report as FlowCal TFX file. Otherwise send as plain text.
<b>TFX/TSV</b> If the report supports it, send report as FlowCal TFX file. Otherwise send as TSV file.</td></tr>
</table>

Send Test File	This option allows the user to test the FTP settings. It attempts to connect to the FTP server, login and transfer a test text file. A dialog box opens up to show the progress of the test. 

FTP Test

```
Sending Test File ...
Connecting to server 66.220.9.51, port 21
220 Welcome to the most popular FTP hosting service! Save on hardware, sof
USER mpfj
331 User name ok, need password.
PASS ********
230 User mpfj logged on.
CWD /mpfj
250 CWD command successful. "/mpfj" is current directory.
TYPE I
200 Type set to I
PASV
227 Entering Passive Mode (66,220,9,51,40,173).
STOR ftptest-20201215093249.txt
150 Connection accepted
226 Transfer complete
QUIT
221 Bye
```

11.5.6 Report Routing

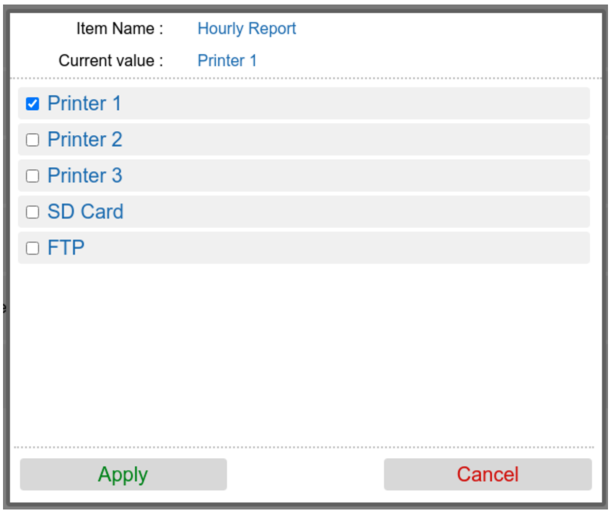
The Report Routing page allows any application reports to be routed to any of the configured printers, SD Card or FTP.

Each report is shown with its current routing selection.

The first item is always “Web Printer” which selects the routing for the Print icon seen on several of the system web pages.

Clicking on the Edit icon opens up a dialog bow showing each available routing destination and a tick box per item. Tick the destinations as required and click Apply.

When a report is then generated, the report is sent to each of the selected destinations.

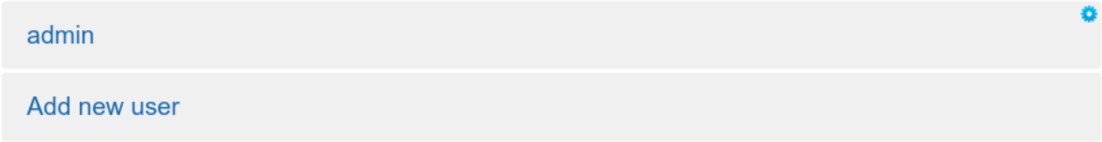


Item Name : Hourly Report
Current value : Printer 1

☑ Printer 1
☐ Printer 2
☐ Printer 3
☐ SD Card
☐ FTP

11.5.7 User Info

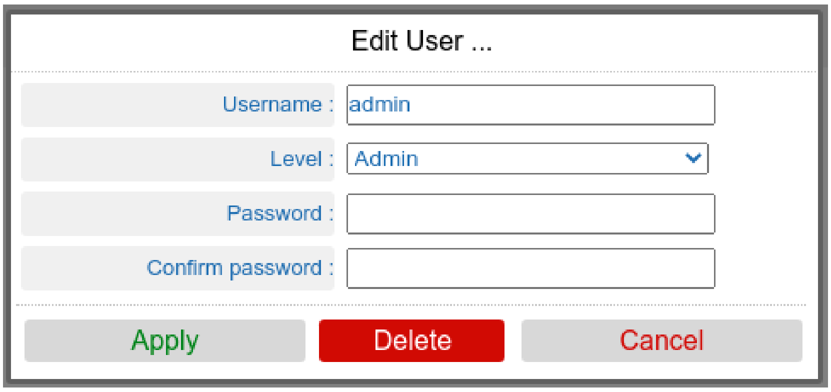
The User Info page shows a list of currently configured users.



admin

Add new user

User details can be viewed by clicking on the relevant Edit icon.



Edit User ...

Username : admin

Level : Admin

Password :

Confirm password :

The following details can be changed:-

Username	Sets the username. All usernames must be unique.
Level	Sets the access level of the user. Six access levels are available, and the names of the access levels are defined by the application via the “USERS” table (see 6.14 - Tables).
Password	To change the password, enter the new password into both password boxes.

Administrator users can edit any user's details or delete a user.

Administrator users can also add a new user by clicking on the “Add new user ...” item and enter the required user details.

Non-administrator users can only edit their own username or password.

11.5.8 Analog Inputs

If the application is configured to use analog inputs (and/or RTD inputs), their current values are shown here.

Clicking on the Edit icon allows the user to perform in-line calibration of the analog inputs.

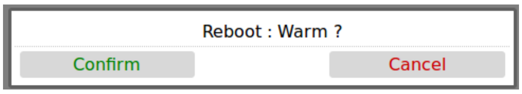
During calibration, a low and high scale can be set. These are only implemented upon acceptance.

A “default” option is also present, which removes any previous in-line calibration values, and returns the relevant input to using its raw values. A prefix asterisk (*) denotes that the raw values are in use.

11.5.9 Reboot

The Reboot page allows administrator users to reboot the target remotely.

Clicking on one of the reboot options will bring up a confirmation dialog to prevent accidental rebooting.



Reboot : Warm ?

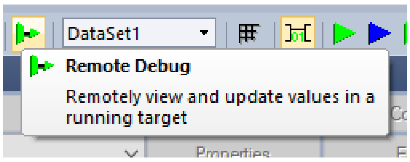
The following reboot options are available:-

Warm	Restarts the target retaining all current persistent pin values.
Cold	Restarts the target resetting all persistent pins back to their default value (as defined by the application).
Cold (Keep Totals)	Similar to a Cold reboot, but all persistent pins contained in a group called “TOTALS” will have their values retained. All other persistent pins will be reset back to their default value (as defined by the application).
Defaults	As Cold reboot, but both network ports are reset to factory default.

12 Remote Debug

Once an application has been installed into a target, it is possible to remotely connect to the target and debug the application live on the target.

Click on the Remote Debug icon (or right-click in the Remote tab and select Remote Debug or select File | Remote Debug a Target...).



Remotely view and update values in a running target

A pop-up dialog will prompt the user for the IP address of the target and the application password. Enter these and then click OK.

The target will be contacted and a copy of the application (contained within the target) will be downloaded into the Visual C||Cure editor.

At this point, the editor can be used to drill down into the application, view the live values of all data pins, and even change data values on-the-fly.

Any data changes will be sent back to the target.

Note that the data values displayed are not those calculated by Visual C||Cure on the PC, rather they are the live values taken from the remote target's database itself.

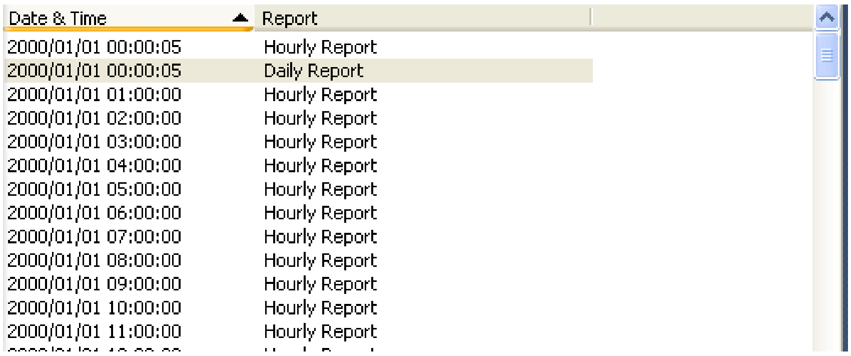
One very powerful feature of the Remote Debug engine is the ability to Single Step codeblocks on the remote target.

If the user enters a codeblock and selects Single Step, the editor will perform the standard Single Step functions as outlined in 5.5.2 - Codeblock Single Step.

Clicking on the Remote Debug icon (or right-click in the Remote tab and select Remote Debug or select File | Remote Debug a Target...) will disconnect the Remote Debug link.

12.1 Remote Reports

When in remote debug mode, the Remote tab also shows the list of available reports that are currently stored on the target.

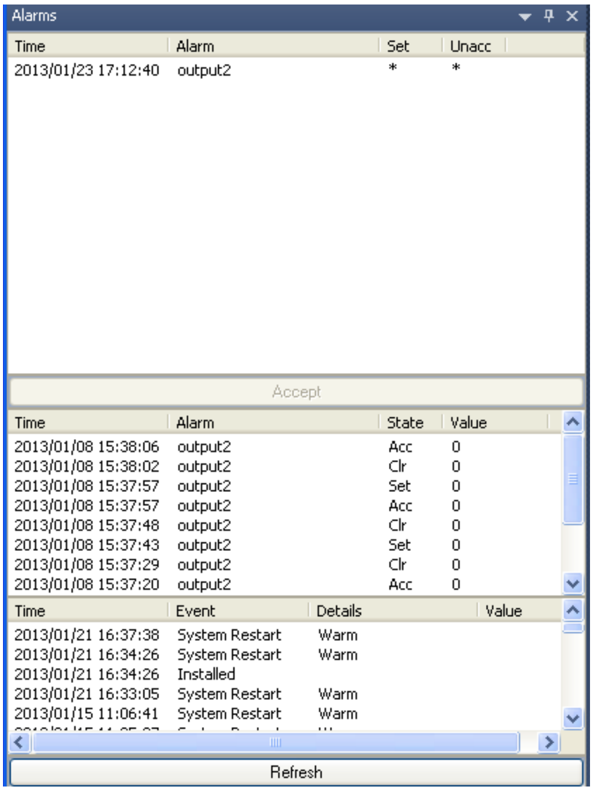


Date & Time	Report
2000/01/01 00:00:05	Hourly Report
2000/01/01 00:00:05	Daily Report
2000/01/01 01:00:00	Hourly Report
2000/01/01 02:00:00	Hourly Report
2000/01/01 03:00:00	Hourly Report
2000/01/01 04:00:00	Hourly Report
2000/01/01 05:00:00	Hourly Report
2000/01/01 06:00:00	Hourly Report
2000/01/01 07:00:00	Hourly Report
2000/01/01 08:00:00	Hourly Report
2000/01/01 09:00:00	Hourly Report
2000/01/01 10:00:00	Hourly Report
2000/01/01 11:00:00	Hourly Report

Double-clicking on any of these reports will load the report data and display the report in the main editor window.

12.2 Remote Alarms / Events

In remote debug mode, the Alarms tab shows a list of the application alarms (and whether they are SET and/or UNACCEPTED), along with a log of all the alarms and event entries stored on the target.



Time	Alarm	Set	Unacc
2013/01/23 17:12:40	output2	*	*

Time	Alarm	State	Value
2013/01/08 15:38:06	output2	Acc	0
2013/01/08 15:38:02	output2	Clr	0
2013/01/08 15:37:57	output2	Set	0
2013/01/08 15:37:57	output2	Acc	0
2013/01/08 15:37:48	output2	Clr	0
2013/01/08 15:37:43	output2	Set	0
2013/01/08 15:37:29	output2	Clr	0
2013/01/08 15:37:20	output2	Acc	0

Time	Event	Details	Value
2013/01/21 16:37:38	System Restart	Warm	
2013/01/21 16:34:26	System Restart	Warm	
2013/01/21 16:34:26	Installed		
2013/01/21 16:33:05	System Restart	Warm	
2013/01/15 11:06:41	System Restart	Warm	

Double-clicking on any of the alarms or alarm log entries will locate and display the relevant alarm pin in the editor window.

This assists the user in quickly diagnosing the cause of any unexpected alarms.

Alarms can be accepted by selecting the relevant alarm and clicking on the Accept button at the bottom of the alarm list.

The Refresh button can be used to query the target for up-to-date Alarm / Event data.

13 Backing Up Applications

Visual C||Cure allows the user to retrieve an application and setup configuration of a target.

Right-click in the Remote tab and select Get a Backup of the Running Application.

A pop-up dialog will prompt the user for the IP address of the target and the target admin password. Enter these and then click OK.

The target will be contacted and the application and current setup will be extracted.

The user will then be prompted for a filename to save the extracted application.

The saved application will contain:-

- The original application as installed on the target
- A copy of the persistent pin values
- A copy of the current system setup (e.g. time/date settings, users, printer settings)

14 Updating Firmware

To assist the long term support provided for the targets, it is possible to update both the system firmware and the I/O firmware remotely.

Update files may be provided by your supplier.

To install such update files, right-click in the Remote tab and select either Update Firmware (to update the system firmware) or Update I/O Firmware.

A pop-up dialog will prompt the user for the IP address of the target and the target admin password. Enter these and then click OK.

The user will then be prompted to select the relevant update file.

Once selected, Visual C||Cure will:-

1. Connect to the target
2. Stop any existing application
3. Transfer the firmware update
4. Restart the target

The update progress will be shown in the App column text on the Remote tab.