

---

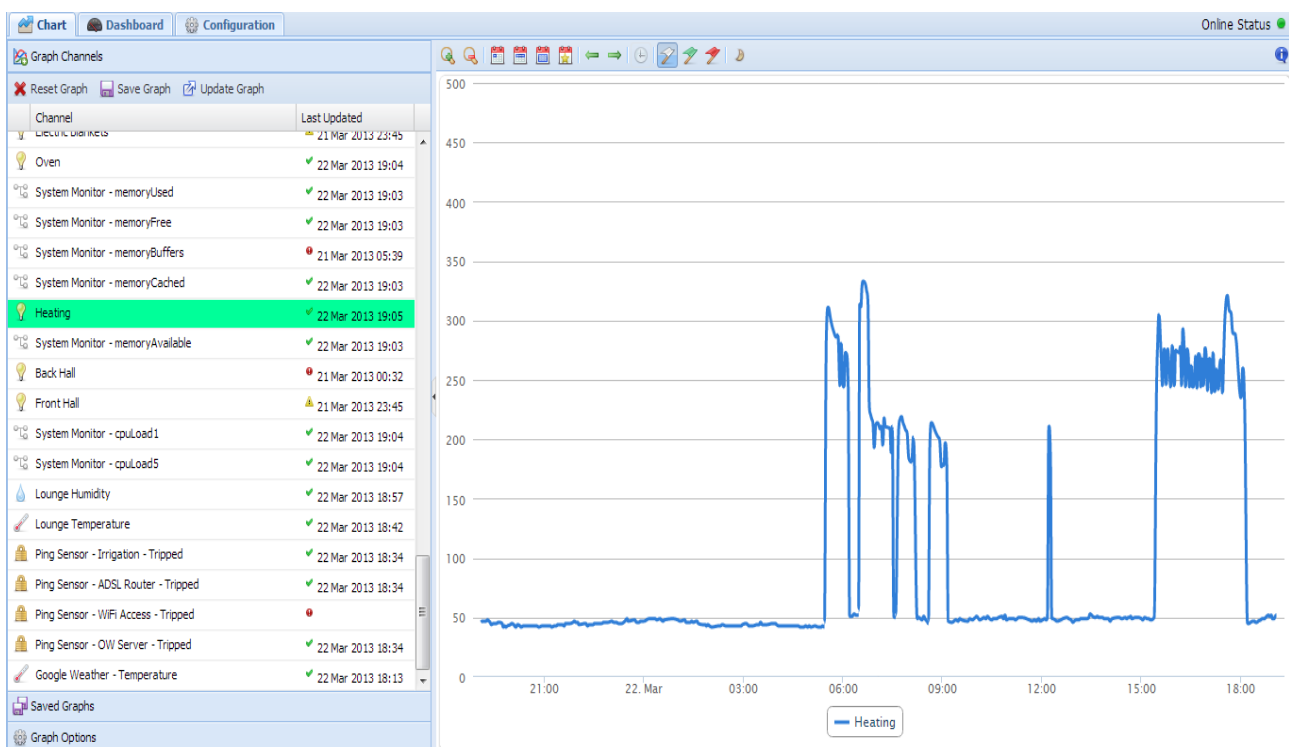
## dataMine Overview

---

*dataMine* is a data capture and visualisation plugin for Vera. It includes a Vera plugin to record and process the data, and a web GUI for data visualisation and configuration. The web GUI is shown in Figure 1.

The plugin that runs in Lua records all changes for variables that are being logged. The raw data is then written into log files on Vera (preferably on a USB stick), and the GUI then requests this data from the plugin. Both sides are therefore required to operate – the Vera, providing the data capture and server, and the GUI, providing the data manipulation and visualisation.

This version of the guide was updated on 08 May 2013.



*dataMine* works by monitoring all changes in device variables in Vera. Whenever a variable changes, *dataMine* gets called, and it logs this data into a file for later use. The GUI then interacts with this data to provide the user a visualisation interface.

---

## Installation

---

*dataMine* is installed through the UI5 marketplace. This installs all required files onto your Vera, however some configuration is required before *dataMine* will run.

A USB stick is highly recommended to store *dataMine* data. Depending on the number of variables you are logging, *dataMine* can generate a reasonable amount of data, additionally, as this data is written regularly, this may cause the internal Vera memory problems if it is used for an extended duration.

The USB stick can be formatted on Windows and plugged into a free USB slot on the Vera. You should avoid formatting the drive with the FAT16 file system as this has limitations. On any modern version of Windows, it should compile as FAT32 be default so this shouldn't be a problem. Linux etx2/ext3 file systems are fine. After formatting in Windows, you should set the disk label in the disk properties – this allows you to identify it once it is plugged into Vera.

You should then run the GUI and you will be presented with the dialogue shown in Figure 1. This dialogue asks you to select the UUID (the Universally Unique Identifier) of the USB to be used for *dataMine*. If you have more than 1 USB stick installed in your Vera, and you're unsure which one is the one for *dataMine*, then it is suggested to add a label to the drive you wish to use for *dataMine*, and refresh the GUI page so that only one UUID is shown.

Once you've selected the UUID, press Save, and *dataMine* should be configured. Restart Luup (press Reload in U15), and *dataMine* should be up and running.

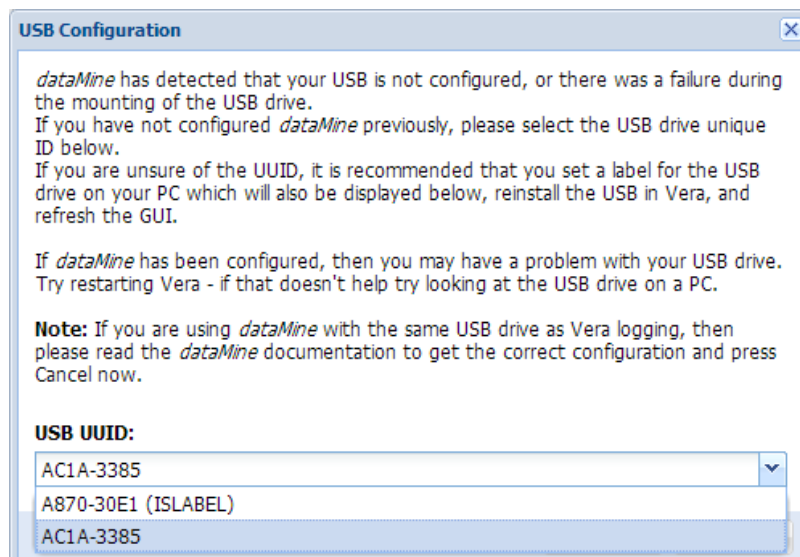


Figure 2: *dataMine* initialisation dialogue.

The configuration dialogue will set the variables shown in Table 1. It is also possible to set these values manually if you please, however the dialogue will be shown anytime that the USB stick fails to mount correctly.

Configuration Variable	Suggested Value
SetDataDirectory	/dataMine
SetMountPoint	Blank
SetManualMount	0
SetMountUUID	USB UUID

Table 1: Suggested configuration for use with Vera USB logging.

When *dataMine* starts, it will create two further variables - "mountLocation" and "mountType". It should be noted that these just report the current mount status, and are not meant to be used as data entry points by the user. If you have manually set a mount point with the "SetMountPoint" setting, then if the mounting works then "mountLocation" will be the same as "SetMountPoint". "mountType" will reflect the filesystem used by the USB stick – typically "vfat" when purchased straight from the shop. This should look similar to what is shown in Figure 3.

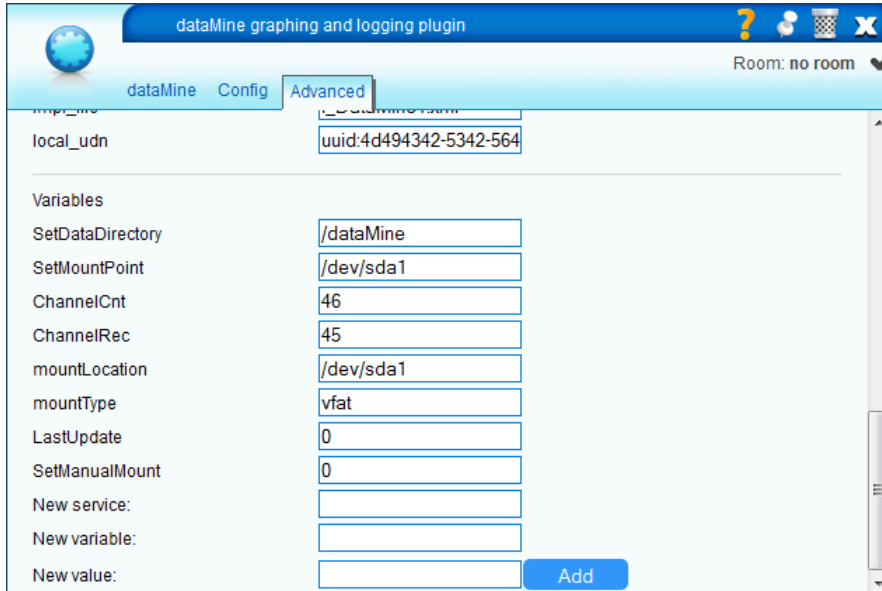


Figure 3: *dataMine* configuration variables.

## Using *dataMine* with the Vera USB logging

For use with the Vera USB logging, additional configuration may be required. This is because Vera will mount the USB stick, and if *dataMine* is also allowed to mount the same stick to a different mount point, corruption can occur. You must either use a separate USB stick for *dataMine* and use the configuration described above, or point the *dataMine* directory to a location below the log directory. Note that the preferred option is to use a separate stick if possible to avoid conflicts.

Suggested configuration for use with USB logging is shown in Table 2.

Configuration Variable	Suggested Value
SetDataDirectory	/tmp/log/cmh/dataMine
SetMountPoint	Not required to be set
SetManualMount	1
SetMountUUID	NA

Table 2: Suggested configuration for use with Vera USB logging.

---

# Use

## *Plugin Properties*

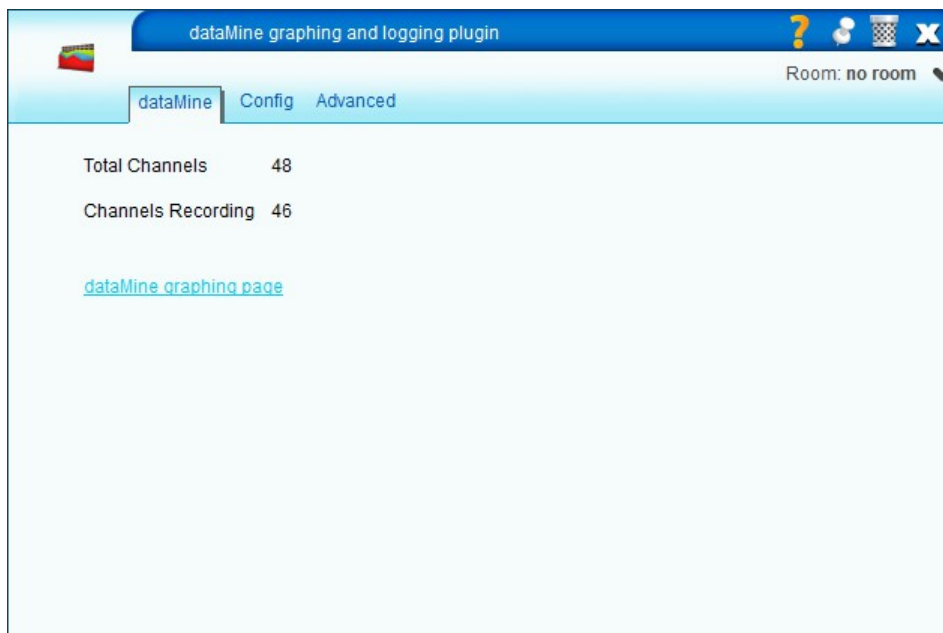
---

The dataMine plugin presents a simple device in the UI5 dashboard – as shown in Figure 4. This just shows how many channels are being monitored, and how many channels have been configured in all.



*Figure 4: Device overview.*

Figure 5 shows the main properties tab. Again, this present the same information as the device overview, however there is also a link to the GUI webpage which is outside of the UI5 interface.



*Figure 5: dataMine properties page.*

---

## Graphical User Interface

---

The GUI is a web interface that allows you to visualise the data from Vera. It uses a tabbed interface to select different data views or types as shown in Figure 6. At the right side of the tab bar is an “Online Status” indicator – this is green ● if the web interface is communicating with Vera, and grey ● if Vera is not communicating with the browser. It displays red ● if communication is ok, however there are errors within *dataMine* (eg. Initialisation error, or error writing data to the USB).

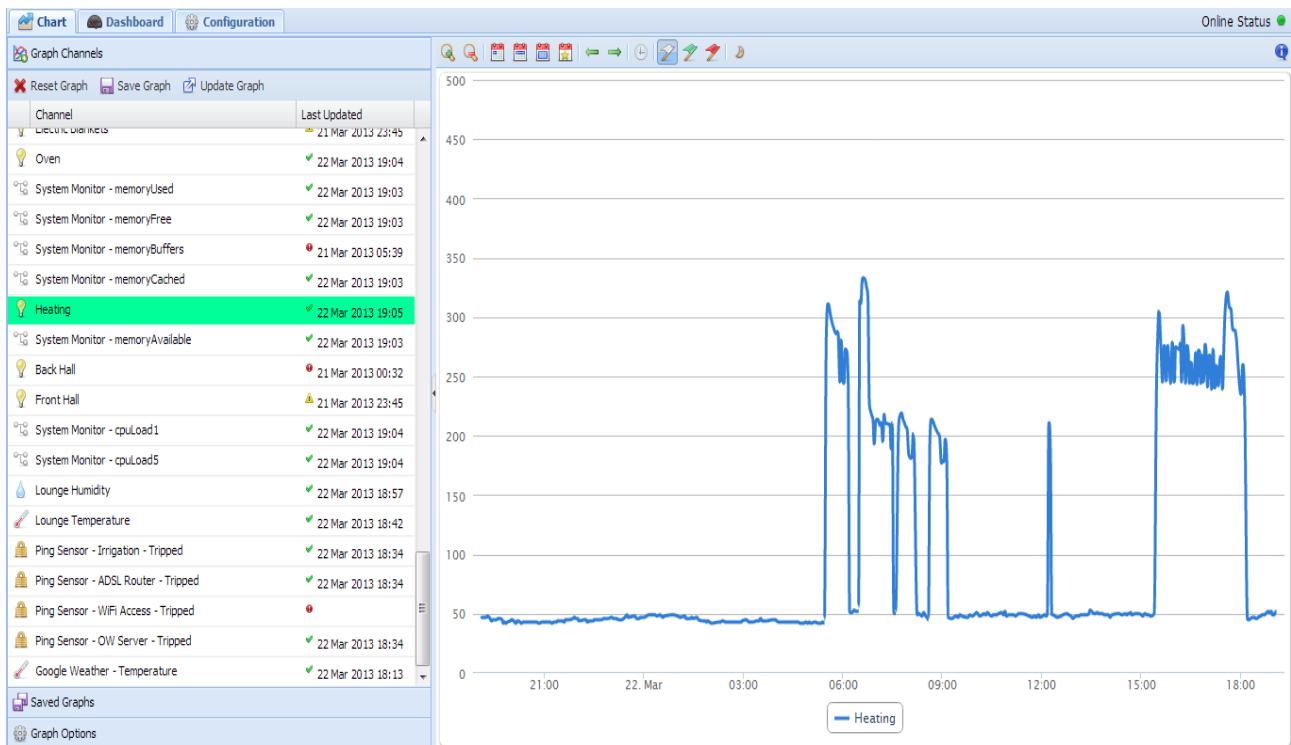


Figure 6: dataMine User Interface.

---

## Graph Tab

---

The graph tab provides an interface to graph the data being recorded by the *dataMine* plugin.

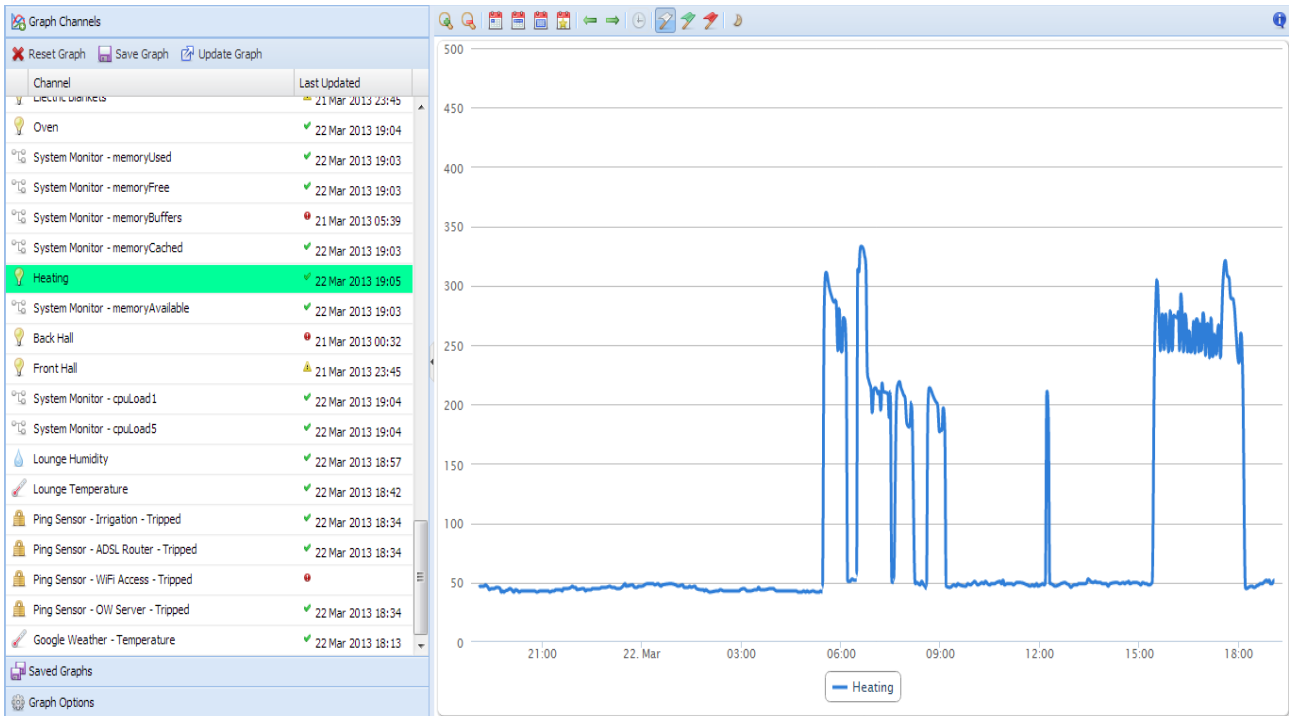


Figure 7: dataMine graph window.

The graph display is split into two main areas – the configuration area to the left, and the graph itself. In between the two is a slider bar that allows adjustment of the position between the two. Note that there's also a button in the middle of this slider with an arrow pointing to the left. Clicking on this will hide the configuration area and you'll have a screen that looks like Figure 8. Clicking on the bar on the left will temporarily display the configuration area, while clicking on the arrows at the top of the bar will restore the configuration area. This allows the full window area to be made available for graphs.

If running dataMine on a mobile browser, the bar will default to the left to provide the most space possible for small screens.

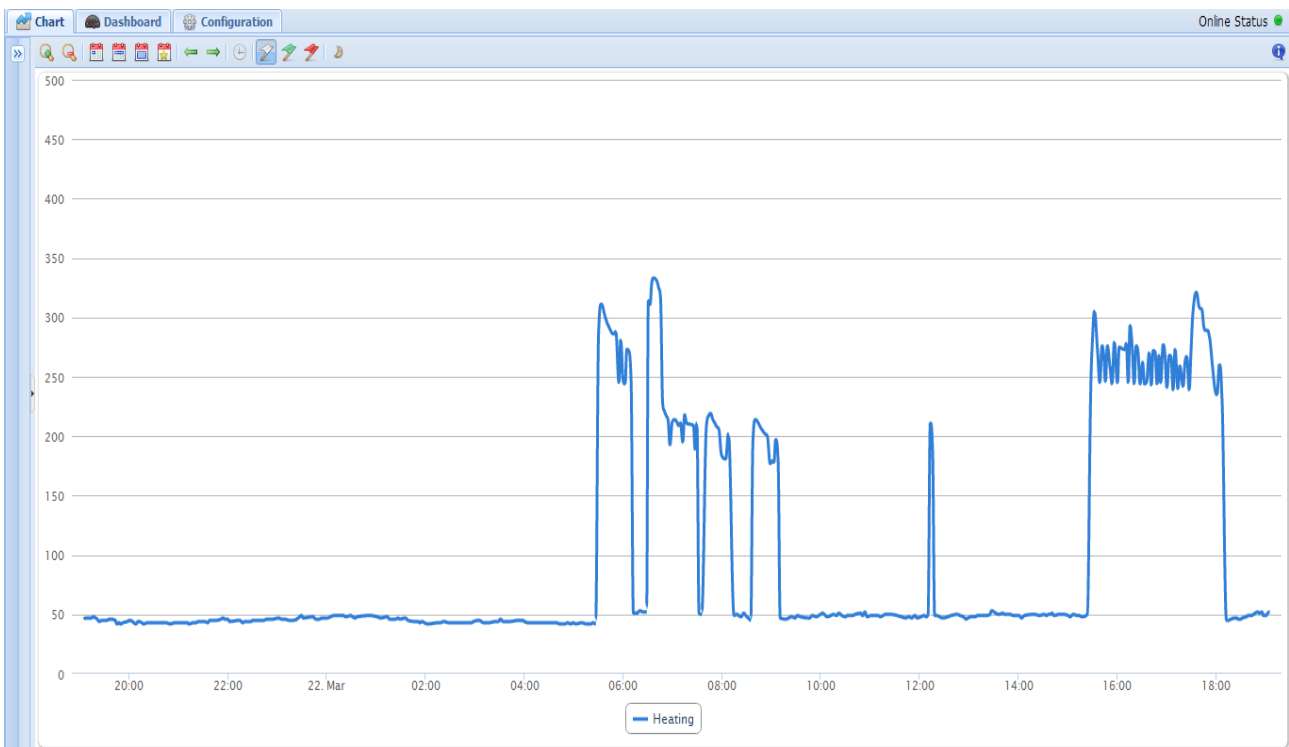






Figure 8: Graph tab with configuration area hidden.

The configuration area includes areas for selecting the channels to graph, and the graph options. In the Graph Channels area, clicking on a variable name, the variable will select or deselect. When selected, the variables can be graphed by clicking on the  *Update Graph* button (the graph will use the same time period as is on the current graph). Clicking on the  *Reset Graph* button will clear the current selection and reset the time period back to default – it will not affect the current graph, however. Clicking on the  *Save Graph* button will allow you to save the graph for later quick recall. Saved graphs will be listed in the  *Saved Graphs* list.

Channel	Last Updated
Irrigation Controller - Watts	23 Dec 2012 00:31
Bathroom - Humidity	27 Dec 2012 10:43
Bathroom - Temperature	27 Dec 2012 10:37
Boiler - Water Temperature	27 Dec 2012 10:50
Google Weather - Condition	27 Dec 2012 10:47
Shower Room Temperature - CurrentTemperature	27 Dec 2012 10:41
Shower Room Humidity - CurrentLevel	27 Dec 2012 10:50
Loft Humidity - CurrentLevel	27 Dec 2012 10:50
Loft Temperature - CurrentTemperature	27 Dec 2012 10:47
Outside Humidity - CurrentLevel	27 Dec 2012 10:50
Pressure (Mb) - CurrentPressure	27 Dec 2012 10:44
Outside Temperature - CurrentTemperature	27 Dec 2012 10:44
Outside Light Sensor - CurrentLevel	27 Dec 2012 10:48
Home Energy Monitor - Watts	27 Dec 2012 10:50
Billion ADSL Router - UpstreamRate	
Billion ADSL Router - DownstreamRate	26 Dec 2012 15:26
Billion ADSL Router - UpstreamSNR	26 Dec 2012 15:26
Billion ADSL Router - DownstreamSNR	27 Dec 2012 10:12

Figure 9: Graph channel selection

Each channel has an icon on the left side - this indicates the type of data. The *Last Updated* column also shows if data has been received recently. A tick ✓ is shown if the variable data has changed recently, a yellow exclamation ⚠ is shown if the data is quite old, and a red exclamation ❗ is shown if the data is very old.

The definition of these flags is configurable per channel since different channel types expect different change rates.

Clicking on the ⚙ *Graph Options* navigation bar brings up a number of parameters that can be changed in the graph. See Figure 10 and Table 3 for details of the options.

Graph Options	
Display data markers	false
Display shadows	false
Display legend	true
Display crosshairs	true
Display tooltip	true
Display local times	true
Line width	3















Figure 10: Graph option window.



Option	Description
Display data markers	Displays a point on the graph where each “real” data point exists. Lines are then drawn between the data points to make up the graph.
Display shadows	Displays a border and shadow around the chart,
Display legend	Displays the graph legend showing what data is displayed with each colour.
Display crosshairs	Enables crosshairs.
Display tooltip	Shows the tooltips where the cursor or crosshair is near a data point. The tooltip shows the actual data displayed on the graph.
Display local times	Switches between local time and Universal time.
Line width	Changes the line width of graphs.
Animation	Enable or disable the chart animation when data is drawn.

*Table 3: Graph options.*

The toolbar across the top of the graph screen provides a number of buttons for interacting with the graph. Each of the buttons has a tooltip, so if you put the mouse over the button, you will get a hint as to what the button does.

Icon	Description
	Zoom in
	Zoom out
	Display last days data
	Display last weeks data
	Display last months data
	Display last years data
	Scroll left
	Scroll right
	Display real-time graph
	Turn off displaying Vera Notifications
	Display notifications relevant to the channels being graphed (from the same device, or its parent device)
	Display all notifications – note that this can be overwhelming when long time-scales are displayed.
	Displays grey bands on the chart during night-time.
	Display graph information

*Table 4: Graph toolbar buttons.*

It is possible to zoom using the mouse as shown in Figure 11. Simply hold the mouse button down at the start of the zoom period, move to the end, and release the mouse. New graph data will be loaded.

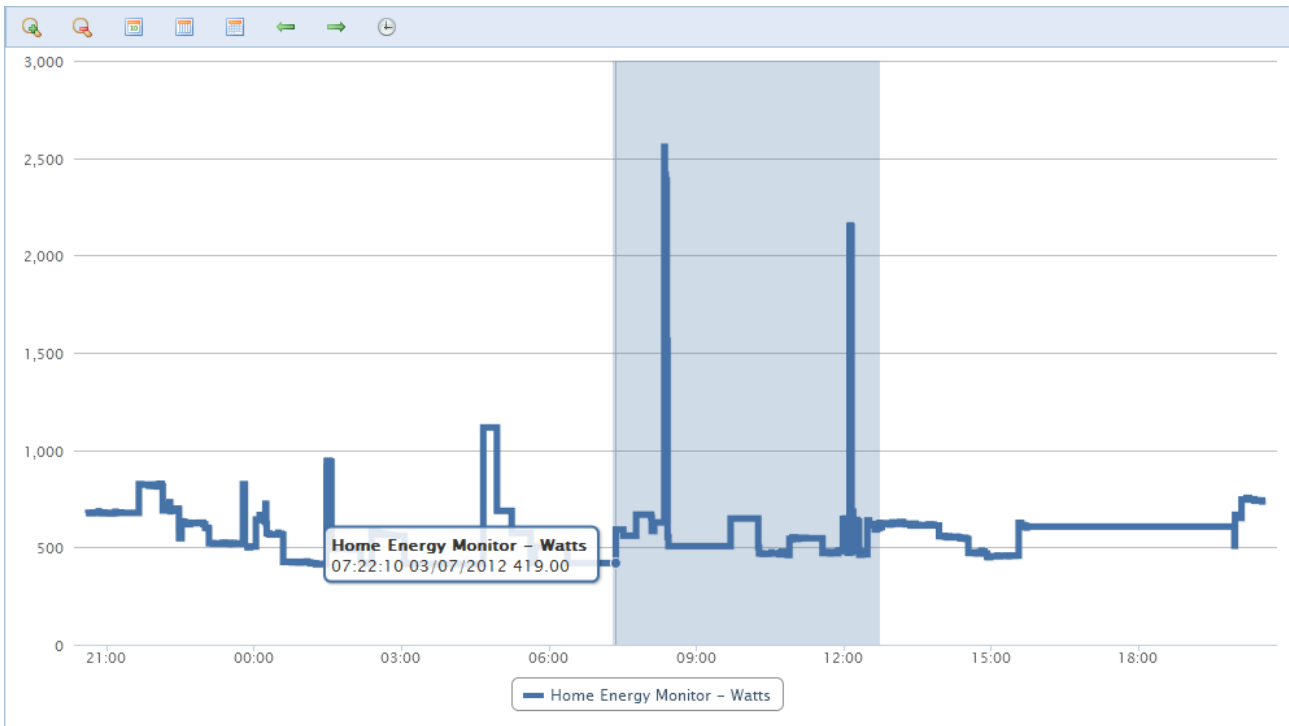


Figure 11: Graph window zoom

The real-time graph displays data for the past 15 minutes, and updates every second. Due to the nature of the display, it displays data in “absolute form” - ie. it does not interpolate between data points. This gives the display a more stepped nature than the standard graph. (Note: Real-time graphing is not implemented in the current version – it will return soon).

## Notifications

*dataMine* can display Vera notifications on the graph as shown in Figure 12. The notifications are updated ever minute, so very new notifications may not be displayed. If there are more than 500 notifications in the graph, *dataMine* will only show a selection of 500 by trimming out every  $n^{\text{th}}$  notification. In reality, this is far too many to be able to view anyway!

Putting your mouse over the notification will display more information on the notification.



Figure 12: Notification display

### Day-Night banding

*dataMine* records the sunrise/sunset times calculated by Vera, and then allows the nights to be displayed as grey bands in the graphs. This is shown in Figure 13. To activate/deactivate this option, click on the moon icon 🌙 in the toolbar.

Note that as *dataMine* does not calculate the sunrise/sunset times, this feature will only work after the date that this was added to *dataMine* (April 2013). Also, if your Vera is turned off for a period, this will also stop the times being logged.

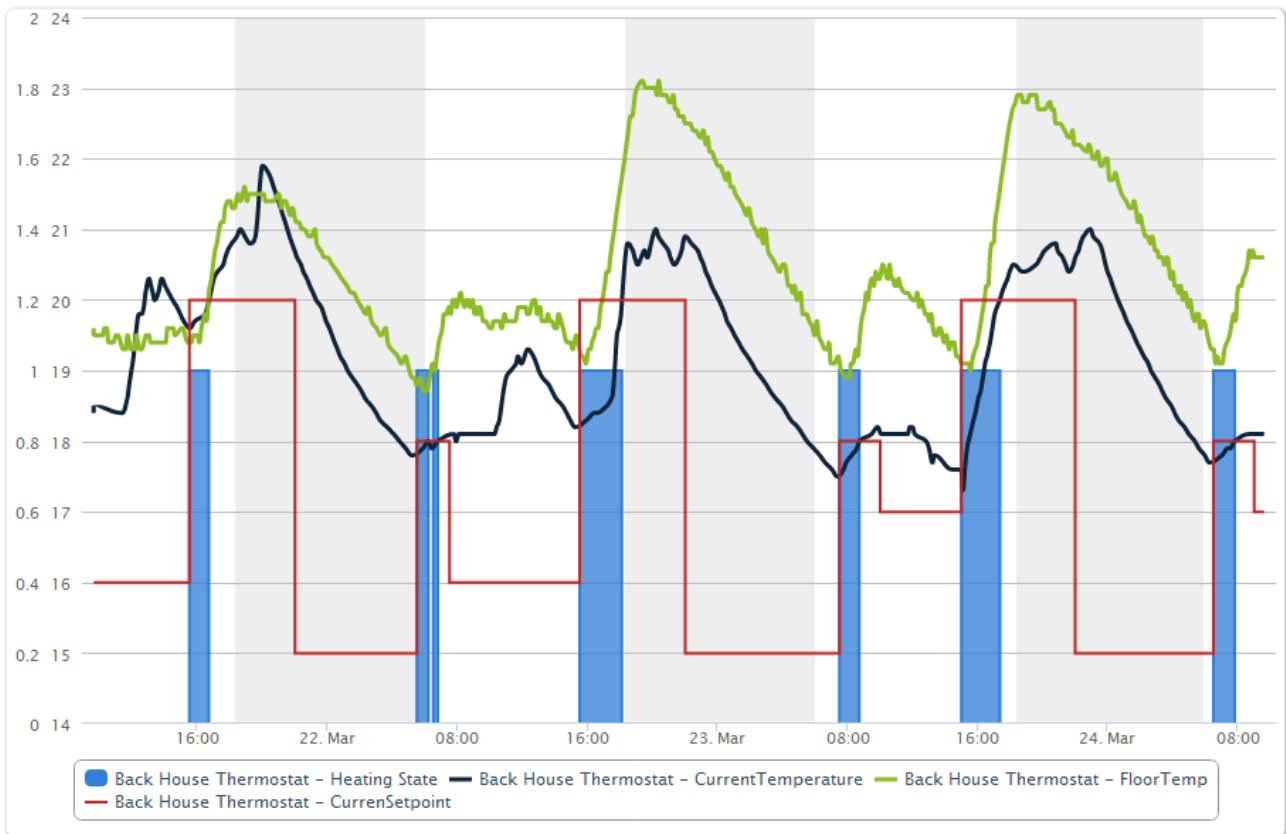


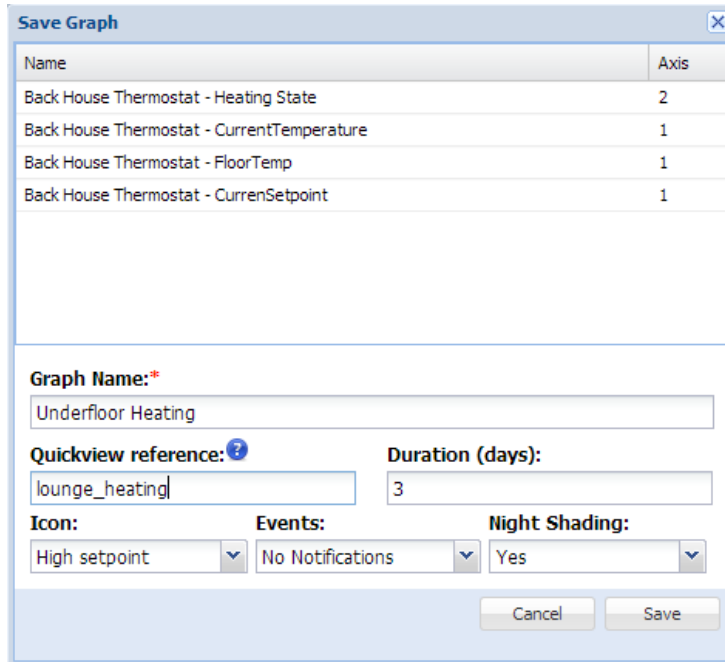
Figure 13: Day-Night banding.

## Saving Graphs

From the graph selection window, you can click the Save button to save a graph configuration for later recall. Figure 14 shows the window that is then displayed. This allows selection of which axis each channel will be displayed on, as well as an icon that will be displayed alongside the graph in the selection list.

The axis selection must be a number between 1 and 4.

The QuickView reference is a reference for displaying the graph in an embedded webpage. For more information, see the section on embedding below. Note that this should NOT contain spaces.

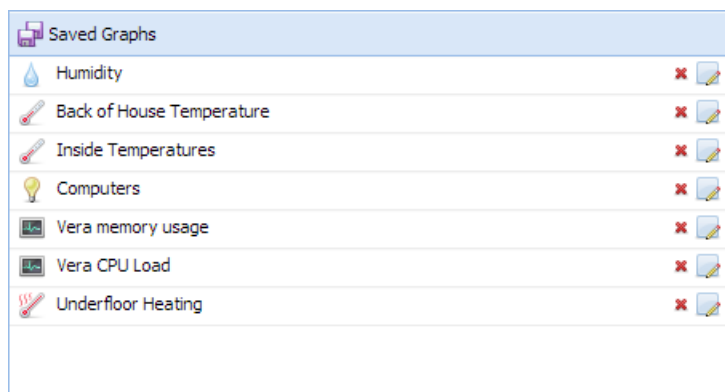


The 'Save Graph' dialog box contains a table with the following data:

Name	Axis
Back House Thermostat - Heating State	2
Back House Thermostat - CurrentTemperature	1
Back House Thermostat - FloorTemp	1
Back House Thermostat - CurrenSetpoint	1

Below the table, the 'Graph Name:' field is set to 'Underfloor Heating'. The 'Quickview reference:' field is 'lounge\_heating' and the 'Duration (days):' field is '3'. The 'Icon:' dropdown is 'High setpoint', 'Events:' is 'No Notifications', and 'Night Shading:' is 'Yes'. 'Cancel' and 'Save' buttons are at the bottom.


Figure 14: Graph Save / Edit window.



The 'Saved Graphs' list contains the following items:

Graph Name	Delete Icon (x)	Edit Icon (pencil)
Humidity	Yes	Yes
Back of House Temperature	Yes	Yes
Inside Temperatures	Yes	Yes
Computers	Yes	Yes
Vera memory usage	Yes	Yes
Vera CPU Load	Yes	Yes
Underfloor Heating	Yes	Yes

Figure 15: Saved graph selection list.

Figure 15 Shows the saved graph selection list. The **x** icon allows you to delete the saved graph, and the  icon allows you to edit the graph data.

Note that when editing a graph, if you change the name, *dataMine* will save a new copy of the graph and you will need to delete the one you don't want.

---

## Dashboard Tab

---

Figure 16 shows the dashboard. Currently this is (very much!) work in progress and it shows the energy currently being used. There is a gauge showing the current whole house energy, and a donut chart showing the split by device and category. A trend chart shows the whole-house use over the past 24 hours.

(This will change).

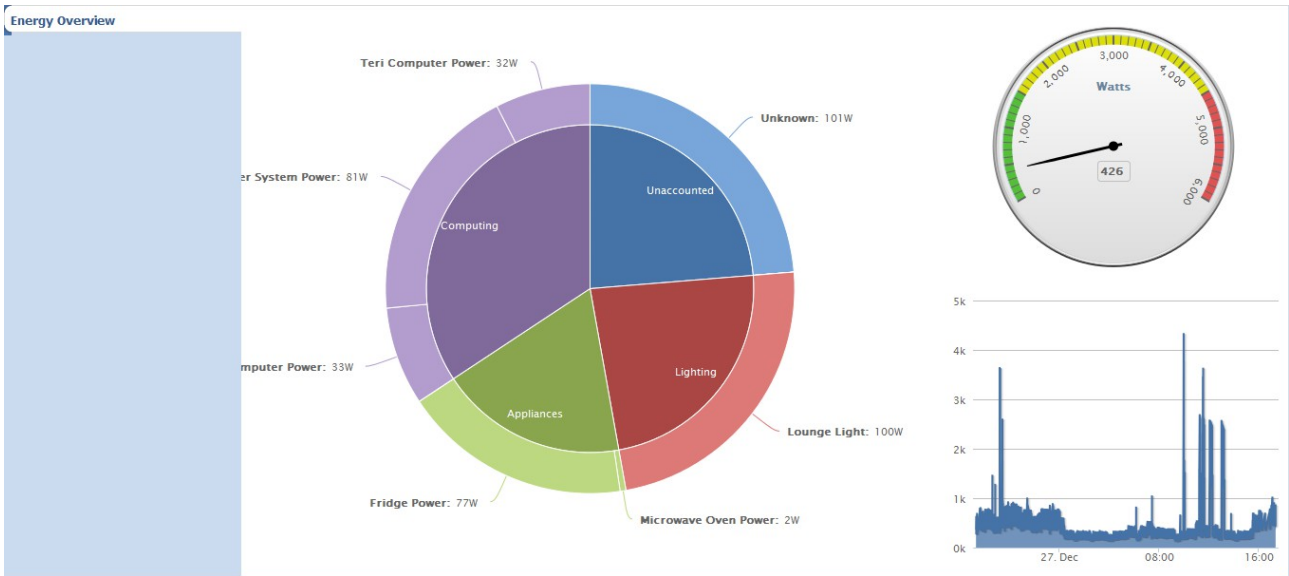
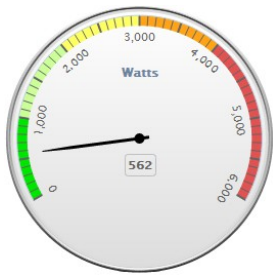
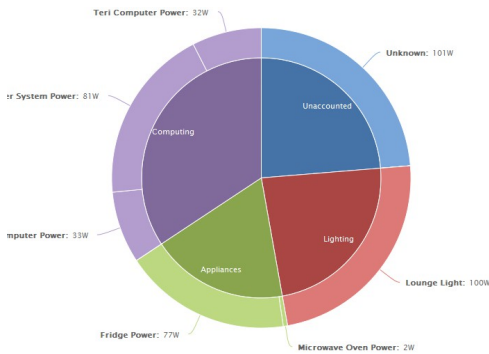


Figure 16: Dashboard tab



The whole-house gauge provides the current whole house energy use. It is a sum of all the whole house energy categories, less the energy generator categories.



The energy consumption donut shows energy consumers by category and device. The Unaccounted category is the difference between the total of all consumers, and the current whole-house consumers (note that this doesn't include the energy generators).

## Configuration Tab

The configuration tab shown in Figure 17 allows configuration of *dataMine*. Currently, this primarily includes the configuration of variable logging, but further configuration will be added in future.

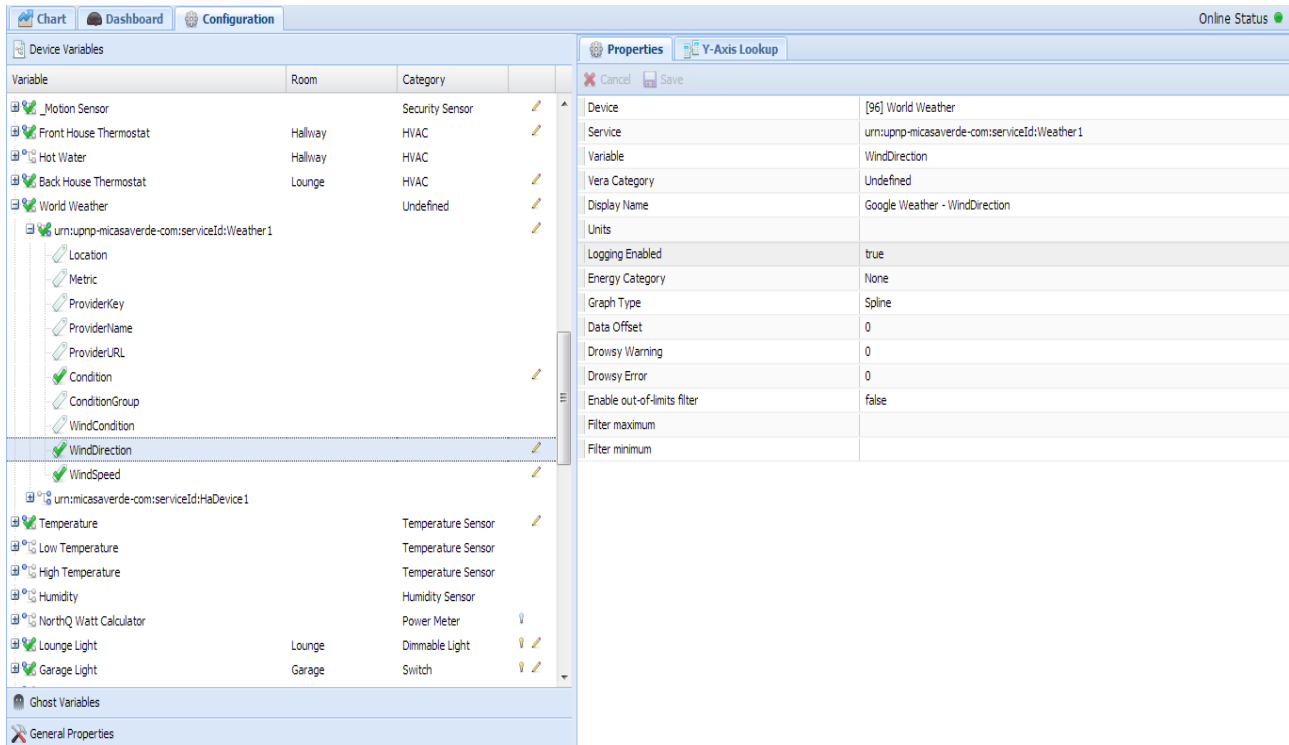



Figure 17: Configuration Tab

## Variable Configuration

Two lists are presented for variable configuration. The active devices are listed under *Device Variables*, and inactive devices are listed under *Ghost Variables*. Ghost variables are variables that still exist in the *dataMine* configuration, but the devices have been deleted from Vera. You can delete the ghost variable, or keep it for reference. Deleting the ghost variable will not delete the historical data from the archive.

Once configuration changes have been made, you must press the  *Save* button for them to take effect.

Option	Description
Device	The Vera device name and number (can not be changed).
Service	The Vera Service name (can not be changed).
Variable	The Vera Variable name (can not be changed).
Vera Category	The Vera Device Category (can not be changed).



Option	Description
Display Name	The display name that will be used in <i>dataMine</i> graphs.
Units	A text string to set the units.
Logging Enabled	Indicates if <i>dataMine</i> is logging data for this variable. Set to <i>true</i> to enable logging.
Energy Category	<p>The <i>dataMine</i> energy category. This is kept separate from the Vera device category. The following categories can be used -:</p> <ul style="list-style-type: none"> <li>1 Whole-house</li> <li>2 Lighting</li> <li>3 Heating</li> <li>4 Entertainment</li> <li>5 Appliances</li> <li>6 Computing</li> <li>7 Utilities</li> <li>21 Solar PV generation</li> <li>22 Wind generation</li> </ul> <p>(Note that this will be made more user friendly in future. It is currently limited by the libraries being used)</p>
Graph Type	<p>Sets the graph type to be used with this channel</p> <ul style="list-style-type: none"> <li>0 Default (spline)</li> <li>1 Line</li> <li>2 Scatter</li> <li>3 Area</li> <li>4 Area/Line combined (for status graphs)</li> </ul> <p>(Note that this will be made more user friendly in future. It is currently limited by the libraries being used)</p>
Data Offset	Adds a data offset when graphing out the channel. The value will be added to each value to allow calibration of a channel. This does not affect the logged data which will always be the value.
Drowsy Warning	Allows modification of the drowsy warning time. This is set in minutes. The drowsy warning is the time after which the ⚠ icon is shown. Setting this to 0 will use the system default.
Drowsy Error	Allows modification of the drowsy error time. This is set in minutes. The drowsy error is the time after which the 🚫 icon is shown. Setting this to 0 will use the system default.
Enable out-of-limits filter	Enables or disables filtering of values. Filtered data outside of the specified limits will not be displayed in graphs. This can be used to filter out large spikes.
Filter maximum	Maximum filter value. Data above this number will not be displayed.
Filter minimum	Minimum filter value. Data below this number will not be displayed.

Table 5: Variable configuration options.

Figure 18 shows the y-axis lookup. This is primarily useful for alphanumeric data (wind direction, weather condition etc). *dataMine* actually generates this lookup automatically for such data, however this configuration window allows you to change the way the data is graphed. By changing the numbers against the labels, you will change the order the data is graphed in the vertical axis.

It is also possible to use this configuration for numerical data that you would prefer to be displayed using labels. Note however that this may produce strange results for data with continually varying data, and is best used for variables with a limited number of possible points (as per the data mentioned above).

Label	Value
WSW	12
ESE	6
NNE	2
SSE	8
SSW	10
WNW	14
East	5
North	1
NE	3
NW	15
West	13
South	9
ENE	4
SW	11
NNW	16
SE	7

*Figure 18: Y-axis lookup configuration*

---

## Embedding graphs in a web page

---

When saving a graph, there is the option to specify a “QuickView reference”. This is a short name for the graph – it must not contain spaces, and it should be kept short. You can then use this reference to embed a full window page into a web browser to display the saved graph.

The URL should be `<vera ip>/dm/graph.html?ref=<quickref>&period=<days>&update=<mins>`

If period is specified, it will override the number of days to display the graph, otherwise it will default to the number of days specified when the graph was saved.

If update is specified, the graph will automatically be updated periodically. The update time is in minutes.

Embedded graphs are meant to be quick and full screen. They do not offer the same interaction as

the full *dataMine* GUI. Day/night boundaries can be displayed, but notifications are not supported.

---

## Advanced Topics

### *Under the hood*

---

*dataMine* works by looking for changes in device variables – it is not periodically logging data based on time, it simply logs changes. This is important to note when thinking about how a graph is rendered – for status type data (eg. on/off), the graph needs to account for this to avoid diagonal lines.

*dataMine* records data in CSV files (Comma Separated Value). (Currently) three different files are generated for each variable being recorded - a raw, hourly and daily file. Files are then split to avoid them getting too large (which is a potential problem for constantly changing sensor data such as temperature). Each time Vera observes a change in the data for a monitored variable, *dataMine* will log the current value into a file. Each day, the long term history files are generated from the raw data.

Each line of data in the CSV logfiles includes two numbers - the time of the data-point (as a unix timestamp), and the data value. For the raw file, the actual changes are logged. For the hourly and daily files, the minimum and maximum value and time are logged for each time period.

---

### *USB and Linux*

---

Linux has a different philosophy to Windows when it comes to attaching USB memory sticks, or disk drives. Such devices are “mounted” into a directory in the root file-system, rather than being seen as a separate drive. A directory is first created in the root file-system, and the USB stick is then mounted into this directory. This has the following implications -:

- Before the USB stick is mounted, if you were to look in the directory, you would see files that are resident in the main file-system.
- After the USB stick is mounted, you will see files that are in the root directory of the USB stick. If there were any files in the mount directory on the main file-system, these will no longer be visible as this directory is effectively replaced by the USB stick.

---

### *Setting system defaults*

---

A UI will be added to set the system defaults, but for now, this can be done in the following way...

Type the following URL into your browser

```
<<Vera Address>>/port_3480/data_request?id=lr_dmCtrl&control=saveConfig&
```

then add the following parameter to the end of the above line to set variables (note – configuration names are case sensitive!).

energyWholeHouseMin=num	Sets the minimum value of the whole house gauge Can not be set above 0.
energyWholeHouseMax=num	Sets the maximum value of the whole house gauge. Can not be set less than 0.
generalDrowsyWarning=num	Sets the drowsy warning system default. Values in milliseconds – defaults to 1 hour (3600000)
generalDrowsyError=num	Sets the drowsy error system default. Values in milliseconds – defaults to 1 day (86400000)
graphDefaultPeriod=num	Sets the default graph period. Values in days – defaults to 1 day. Must be between 1 to 30.

You should get the “OK” response if everything went ok. Note that no error checking is performed when you make changes – these settings are used in the GUI, and when the GUI reads them back from Vera, it does perform a sanity check and may reset to a default if they are deemed to be outside of normal bounds.

---

## **Backing up dataMine data**

---

The following is a method from @vespaman on how to automatically keep a backup of your dataMine data directory. This runs on linux. Note that you need to change the ip address and directories.

From @vespaman original post January 06, 2013  
(<http://forum.micasaverde.com/index.php/topic,13078.msg97263.html#msg97263>)

OK, so I have now whipped something up, which \*seems\* to do about what I need for now. 😊  
I think, that if the logs get very big over time, maybe this isn't the correct method. But at least it can run at times where house is at sleep.

Basically three steps:

1. Create public ssh keys for unattended login to your Vera from your linux server
2. Using scp script, copy the whole dataMine log direcorey to your linux server.
3. Put a line in your crontab so it is done once every night.

It would have been nice to use rsync instead of scp, but it seems that it is not included in the vera dist, so scp will have to do. Not sure how much data we will see in the end, but time will tell.

So in detail for anyone that wants to use this type of backup:

1. Create your public keys according to e.g. this guide: <http://wiki.openwrt.org/doku.php?id=oldwiki:dropbearpublickeyauthenticationhowto>

2. Modify the script for your path's and vera ip (if you don't have 'vera' in your dns/hosts file). Along with your veras IP, you will want to change both destdir and the source path on your vera.

```
#!/bin/bash
date=`date +%Y-%m-%d`
olddate=`date --date '2 months ago' +%Y-%m`
destdir=~/.backup/datamine
# get the files. Vera does not have rsync, so we use regular scp
scp -qpr root@vera:/mnt/datamine_logs $destdir/
# create the backup tar ball
tar -cpzf $destdir/datamine_`date`.tar.gz $destdir/datamine_logs
# fix rights..
chmod 600 $destdir/datamine_`date`.tar.gz
# remove tempfiles
rm -rf $destdir/datamine_logs
# Remove all backups created 2 months ago
rm -f $destdir/datamine_`olddate`-*.tar.gz
```

Create the destdir ( in this case ~/.backup/datamine )

3. Once you have tested the above configuration, and you are sure it works just add line to your own crontab (crontab -e in my case) and add a line like this (running 5 o'clock in the morning).

```
0 5 * * *    ~/.bin/backup_vera_datamine
```

---

## ***If you have a problem***

---

If you have a problem, I'm happy to help. However, please send the Vera logfiles, provide information on what Vera model you are using, if you have other USB sticks fitted.

There is now a quick way to get some of this with the following URL

```
<<VERA IP>>/port_3480/data_request?id=lr_dmCtrl&control=debug
```

This will dump your configuration, USB mount data, and any recent log entries. However, this will only work once the basic plugin is loaded and running. Prior to that, please provide as much information as you can.

### **USB: Read-only file system**

```
Logging error logging variable 67::urn:upnp-org:serviceId:TemperatureSensor1::CurrentTemperature
Unable to open file for write /dataMine/Loft Temperature - CurrentTemperature [R2249].txt
Error: '/dataMine/Loft Temperature - CurrentTemperature [R2249].txt: Read-only file system'
```

Caused by errors in the file system. Copy all files off and reformat.