

DataMine2 Overview

dataMine2 is a derivative work built on the original dataMine logging and graphing plugin for Vera. It includes a Vera plugin and a web GUI for data visualisation. The web GUI can be accessed via <http://vera-ip/dm> on your local network or it can also be accessed remotely following appropriate authentication (see later).

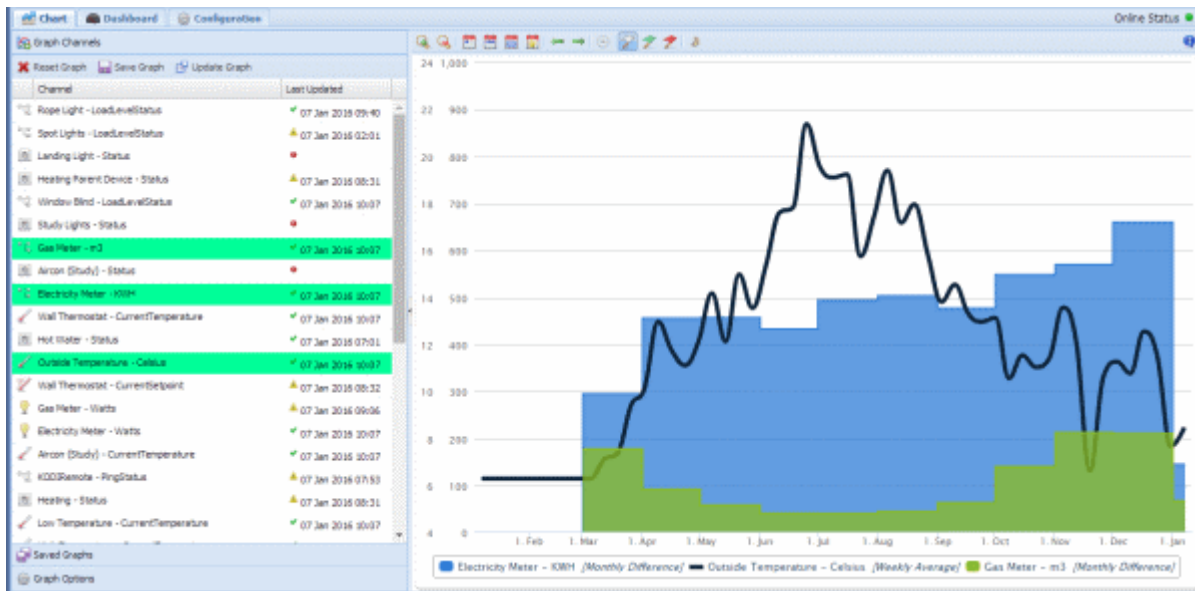


Figure 1. dataMine2 Chart Screen

dataMine2 works by monitoring all changes in device variables in Vera. Whenever a variable changes, dataMine2 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.

New features in dataMine2 include:

- Ability to aggregate data by hour, day, week, month and none (as before)
- Aggregation functions of difference, sum, weighted average, minimum, maximum, on-duration and sample (as before)
- A new API call to provide aggregated data
- A new API call to provide data in a format suitable for Google Charts
- Ability to access web UI remotely
- Various performance improvements

This version of the guide was updated on 11 January 2016 for more up to date information see the discussion forum here: <http://forum.micasaverde.com/index.php/topic,35592.0.html>. Kudos to @Chris who wrote the original code and manual for dataMine , upon which this version is based.

Installation

dataMine2 is installed through the UI5 marketplace (<http://apps.mios.com/plugin.php?id=8526>). This installs all required files onto your Vera, however some configuration is required before dataMine2 will run.

If installed the original dataMine (version 1) must be uninstalled before installing dataMine2 as they use the same web site addresses and will conflict with each other. As dataMine2 also uses the same configuration and log files as dataMine, all your data will be preserved in the new version. However, please make a note of all your advanced properties such as USB mount point shown in Table 1 as they may be useful during installation of dataMine2.

A USB stick is highly recommended to store dataMine2 data. Depending on the number of variables you are logging, dataMine2 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 ext2/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 at <http://vera-ip/dm> 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 UI5), and dataMine should be up and running.

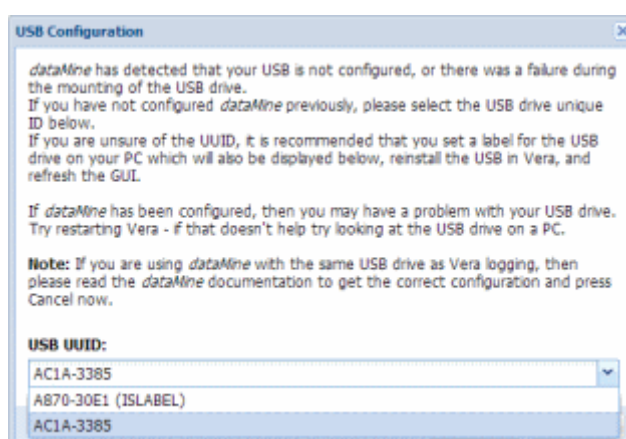


Figure 2. dataMine initialisation dialogue

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.

DataMine2 for Vera Overview and User Guide

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 Table 1.

Variable	Typical Value	Description
errorStatus	false	Indicates if the plug-in is in an error state
SetDataDirectory	/dataMine/	Directory to log data to
SetMountUUID	6A0D-39FD	UUID of any USB stick being logged to
SetMountPoint	<blank>	Manual override for the mount point
SetMountOptions	<blank>	Manual override for the mount options
SetManualMount	0	If set to 1 use above values to mount
SetTimeoutPeriod	12	Time in seconds before a request for graph data times out
SetHistoryEnable	1	No longer used
SetUseBackup	1	If set to 1 backs up old configuration files
SetEventsEnable	0	If set to 1 enables reading of events from the Micasaverde servers. Note this has a big performance impact on graphing and should only be enabled if really necessary.
ChannelCnt	46	Set to the total number of channels configured
ChannelRec	35	Set to the total number of channels being logged
diskTotal	123216	Set to total disk space available for logging (in kB)
diskUsed	4421	Set to total disk space used (in kB)
diskFree	118795	Set to total free disk space (in kB)
diskUsedPcnt	4	Set to percentage disk space used
LastUpdate	0	
mountLocation	/dev/sda1	Set to mount location being used
mountType	vfat	Set to mount file system type

Table 1: Suggested configuration for use with Vera USB logging

dataMine2 was written for Vera UI5 but has been reported to work on UI7 (see forums <http://forum.micasaverde.com/index.php/board,43.0.html> for help on this)

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.

DataMine2 for Vera Overview and User Guide

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

Plugin Properties

The dataMine plugin presents a simple device in the UI5 dashboard. This just shows how many channels have been configured in all. The main properties tab presents further information about disk space and links to the web GUI and debug page which are outside of the UI5 interface. Note that the web GUI can be accessed directly from <http://vera-ip/dm> on your local network or also indirectly via http://vera-ip/port_3480/data_request?id=lr_dmPage – accessing the GUI this way will be slower than the direct route as all files have to be processed to be compatible with remote access.

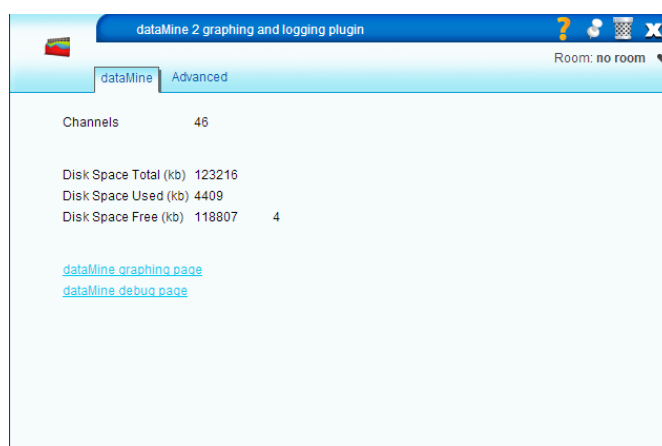


Figure 3. dataMine Properties Page

The best way to access the web GUI remotely is to log-in to the Micasaverde servers (you will need to register first). Login here: <https://cp.mios.com/login.php> (which sets a special authentication cookie) and navigate to your UI5 web portal then click the above *dataMine graphing page* link. Alternatively the GUI can be accessed remotely via a URL of the format:

https://active_server/username/password/serialNumber/data_request?id=lr_dmPage

e.g. https://fwd8.mios.com/username/password/12345678/data_request?id=lr_dmPage

The request must be https (and not http) which will mean that your username and password will be encrypted during transmission. The required values for `active_server` and `serialNumber` can be found via the following request (whilst local to your Vera box):

https://sta1.mios.com/locator_json.php

As UI7 uses a different authentication system to UI5, the above approach may not work. Can anyone help me out with that?

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).

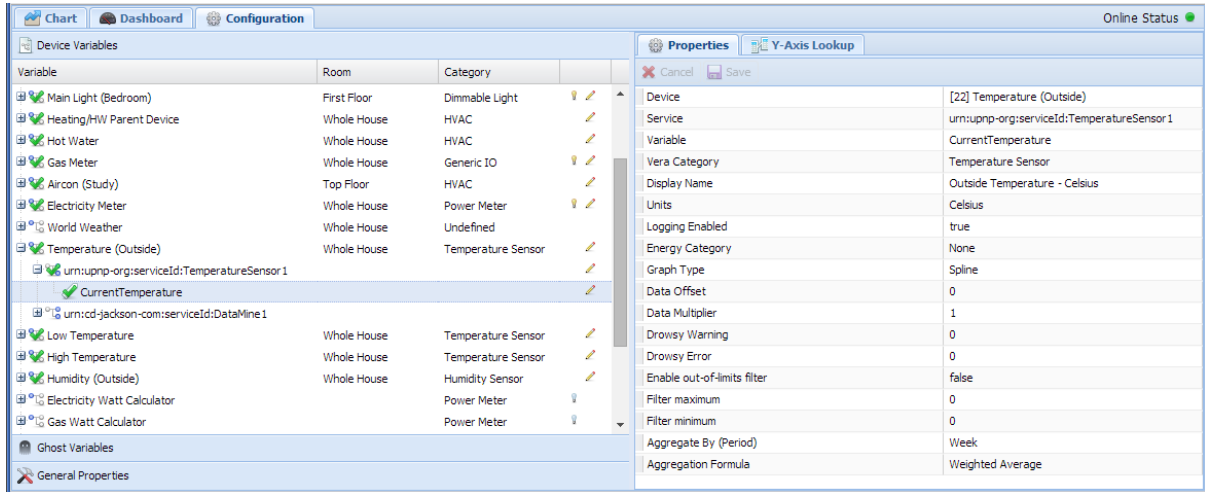


Figure 4. dataMine User Interface

Configuration Tab

Before anything can be graphed, the variables to be watched must first be enabled for logging. To do this click on the Configuration tab and select the device and variable you wish to log in the left-hand Device Variables pane. In the right-hand Properties panel various details will be shown. To enable logging set Logging Enabled to true and click the Save icon. The variable will start logging immediately, but as only *changes* to variables are recorded it may be some time before any data is collected. Furthermore, battery operated devices may only report their status periodically.

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

Variable	Description
Device	Name of device being watched from your Vera dashboard
Service	Service Id of variable being watched
Variable	Name of variable being watched
Vera Category	Category of variable (assigned by Vera)
Display Name	User editable name to be shown on graphs
Units	User editable unit name to be shown on graphs
Logging Enabled	Indicates if dataMine is logging data for this variable. Set to true to enable logging.

DataMine2 for Vera Overview and User Guide

Energy Category	If the variable is the number of watts being generated or consumed by a device an energy category can be set and displayed on the Dashboard tab. The following categories can be used -: 1 Whole-house 2 Lighting 3 Heating 4 Entertainment 5 Appliances 6 Computing 7 Utilities 21 Solar PV generation 22 Wind generation
Graph Type	Spline (default) – interpolated line chart Line – most accurate representation of data collected Scatter – plot a single dot per data point collected Area – fill chart and force Y-axis to start at zero Area/Line – combination chart – plotted as a column chart on certain pages
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.
Data Multiplier	Multiply every data point by this value (before applying the offset)
Drowsy Warning	How long to wait (in minutes) without an update to the watched variable before putting a yellow triangle next to the variable on the Chart tab, Graph Channels pane.
Drowsy Error	How long to wait (in minutes) without an update to the watched variable before putting a red circle next to the variable on the Chart tab, Graph Channels pane.
Enable out-of-limits filter*	Set to true to filter all data points to a minimum and maximum value, which can be used to filter out spikes in the data. Note the limits are filtered before any aggregation calculation which may radically alter the values presented.
Filter maximum	Maximum filter value (if enabled)
Filter minimum	Minimum filter value (if enabled)
Aggregate By (Period)	See below
Aggregation Formula	See below

* In version 1 of dataMine, the filtering process was done on the web page (client side) and so any data requested by the API was unfiltered. In dataMine2 filtering is done server side before the aggregation calculation and so data requested via the API is also filtered.

Configuring Aggregations

dataMine2 only records changes to variables being watched and so there is no need to set any sampling frequency. This has the advantage of being easy to configure and allowing all data to be retained without ever throwing any data away. However, whilst the detail of the data is retained, it can be difficult to see the bigger picture without using aggregations. dataMine2 performs all aggregation calculations at the time of the request without attempting to pre-calculate or alter the underlying data recorded. This saves on storage space but charts that have to trawl through large

DataMine2 for Vera Overview and User Guide

amounts of data may take a few seconds (a typical VeraLite box can aggregate around 8,000 data points per second). Aggregation Period can be set to Auto (default), None (raw data), Hour, Day, Week or Month.

- Auto (default) will cause data to be collected in such a way as to have up to 100 data points displayed on the graph at any time. As the scale of the graph changes the number of data points sampled will change to attempt to keep 100 points on the screen.
- None (raw data) will not aggregate the data and will cause data to be displayed exactly as it has been recorded. Note that a maximum of 2,500 points can be displayed on any chart and any more than this will be truncated. This is useful for data that changes rapidly over a short period but infrequently over a long period such as a light switching on or off.
- Hour – aggregate the data by the hour
- Day – aggregate the data by the day
- Week – aggregate the data over 7 days – note that the week starts on a Thursday on most systems and this cannot be configured
- Month – aggregate by the calendar month – beware that value may be aggregated over 28 to 31 days depending on the month, making comparisons difficult.

Due to daylight saving time, the length of a day, week or month can vary by an hour.

Aggregation formula can be set to Sample (default), Difference (Max - Min), Sum, Weighted Average, Minimum, Maximum and On Duration (Hours)

- Sample (default) – takes a sample of the data at the aggregation period – any values between samples are completely ignored. This is useful for data that doesn't change much over the short term.
- Difference (Max - Min) – finds the maximum and the minimum data values within the aggregation period and calculates the difference. This is useful for calculating the amount something has changed such as the amount of electricity used during the period.
- Sum – this adds up all the data values found during the period. This is useful for counting the number of times an infra-red sensor has been activated during a day, for example.
- Weighted average – this calculates the average of the data values during the period taking into account the amount of time the data point was at that value. E.g. if a value was zero for 8 hours and 1 for 16 hours the weighted average for the day would be 0.667.
- Minimum – the minimum data value found during the aggregation period. This is useful for plotting low temperatures
- Maximum – the maximum data value found during the aggregation period. This is useful for plotting high temperatures
- On Duration (Hours) – the number of hours the variable was greater or equal to 1. This is useful for determining how long a device such as a light switch or heating controller was on during the day.

Once the aggregation value has been set it must be saved. All further graphs generated will use the new aggregation values and none of the underlying data will be changed.

Aggregation Example 1 – Heating On-Time Graph

To create a graph showing the number of hours the heating has been on for during the day log the SwitchPower status (values of 0 or 1) of your heating controller and set the Aggregate By (Period) to "Day" and the Aggregation Formula to "On Duration (Hours)". If your heating controller outputs values other than 0 or 1 (e.g. text values) then use Y-Axis look-up (see below) to translate them appropriately. To make a chart appear as a column type, chose Area/Line in datamine as the Graph Type under the Configuration Properties tab. Area/Line charts will be rendered as column charts on the graph2.html page but as Area/Line charts within the main dataMine2 web interface.

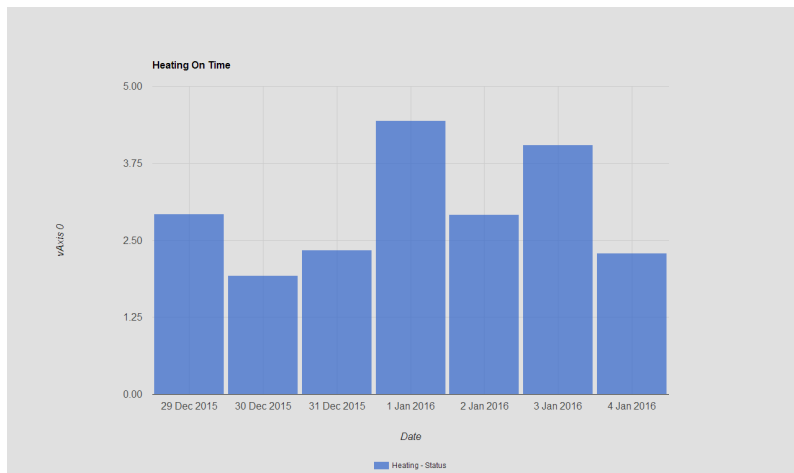


Figure 5. Heating On-Time Graph

Aggregation Example 2 – Gas and Electricity Usage Graph

To create a graph showing your monthly gas and electricity usage, log the number of KWH from your energy monitor and set the Aggregate By (Period) to "Month" and the Aggregation Formula to "Difference (Max - Min)". If your gas meter and electricity meter log in different units (e.g. m³ and KWh) it is possible to do a conversion by setting the Data Multiplier in the configuration. Furthermore, if you want to graph the monthly cost of your gas and electric, you can add a Data Offset value to represent the monthly standing charge and use the Data Multiplier to convert the number of units of gas or electricity to a cost for that period.

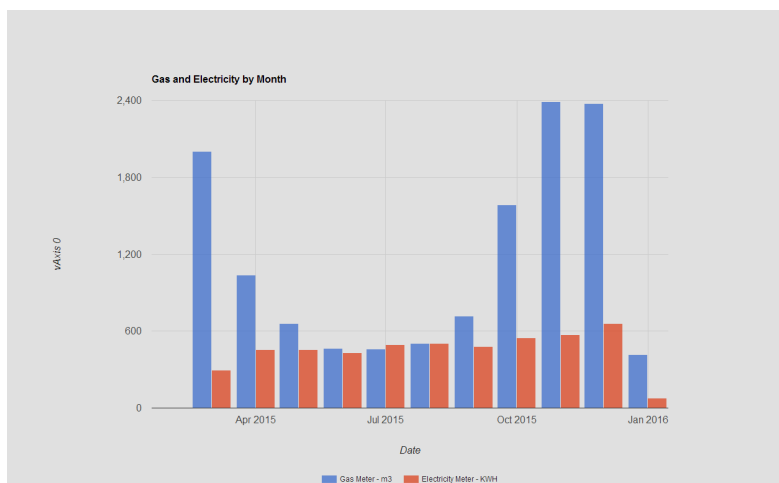
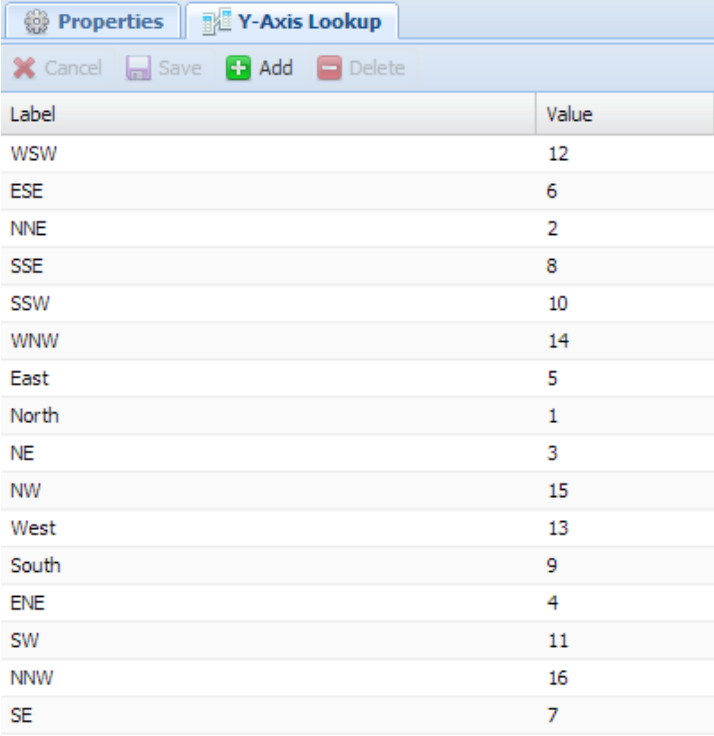


Figure 6. Gas and Electricity Graph

Y-Axis Lookup

Figure below 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 7. Y-axis lookup configuration

Displaying a Graph

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.

DataMine2 for Vera Overview and User Guide

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

To display a graph click on the Chart tab and select a variable in the Channel pane on the left. Items will only show up in this list if they have been enabled for logging on the Configuration tab. The icon on the right shows how long it has been since a value was last recorded. Refer to Drowsy Warning and Drowsy Error to configure when to show the icons.

Multiple channels can be selected simultaneously. Then click Update Graph to show the data and see a graph such as below.

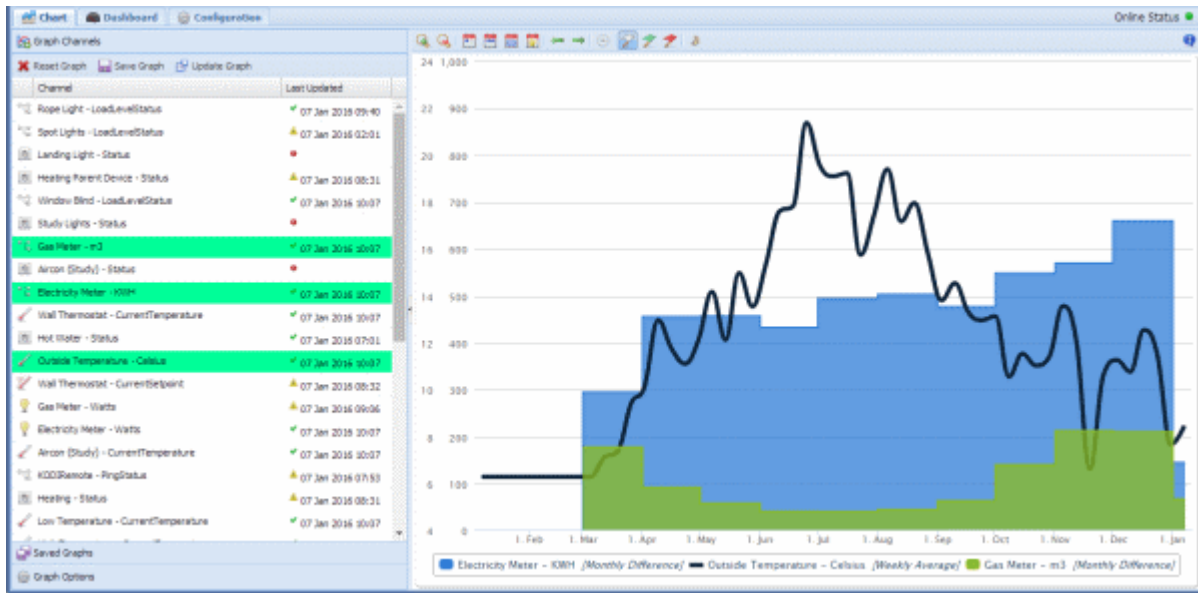

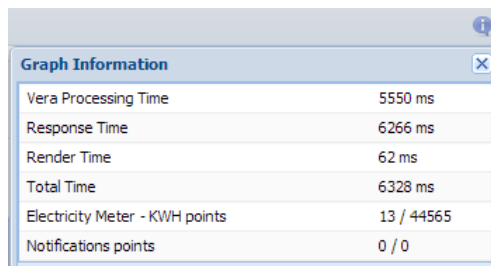


Figure 8. Graph Tab









The variables graphed will be shown at the bottom along with any aggregations applied shown in square brackets. Clicking the  icon in the top right of the main Chart screen shows information about the raw data used to create the graph such as processing time and the number of data points read. In the example below dataMine2 has read 44,565 data points and aggregated them to plot 13 columns on the chart.






Graph Information	
Vera Processing Time	5550 ms
Response Time	6266 ms
Render Time	62 ms
Total Time	6328 ms
Electricity Meter - KWH points	13 / 44565
Notifications points	0 / 0

Figure 9. Graph Information

If Aggregate By is set to anything other than None (raw data) then intermediate data points may be created meaning it is possible to have more output data points plotted than there are data samples.

The   icons can be used to zoom in and out of the data. The     icons can be used to zoom to 1 day, 1 week, 1 month and 1 year of data. The   icons will scroll left and right through

DataMine2 for Vera Overview and User Guide

the data. The  icon shows a real-time graph (not currently working). The  icons enable notifications from the Micasaverde servers to be shown – this will only work if enabled in the advanced settings and has a large performance overhead. Finally  enables or disables showing of night-time bands – these bands are only shown at certain zoom scales and are only available for the prior 90 days.

It is possible to zoom using the mouse as shown in Figure below. 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.

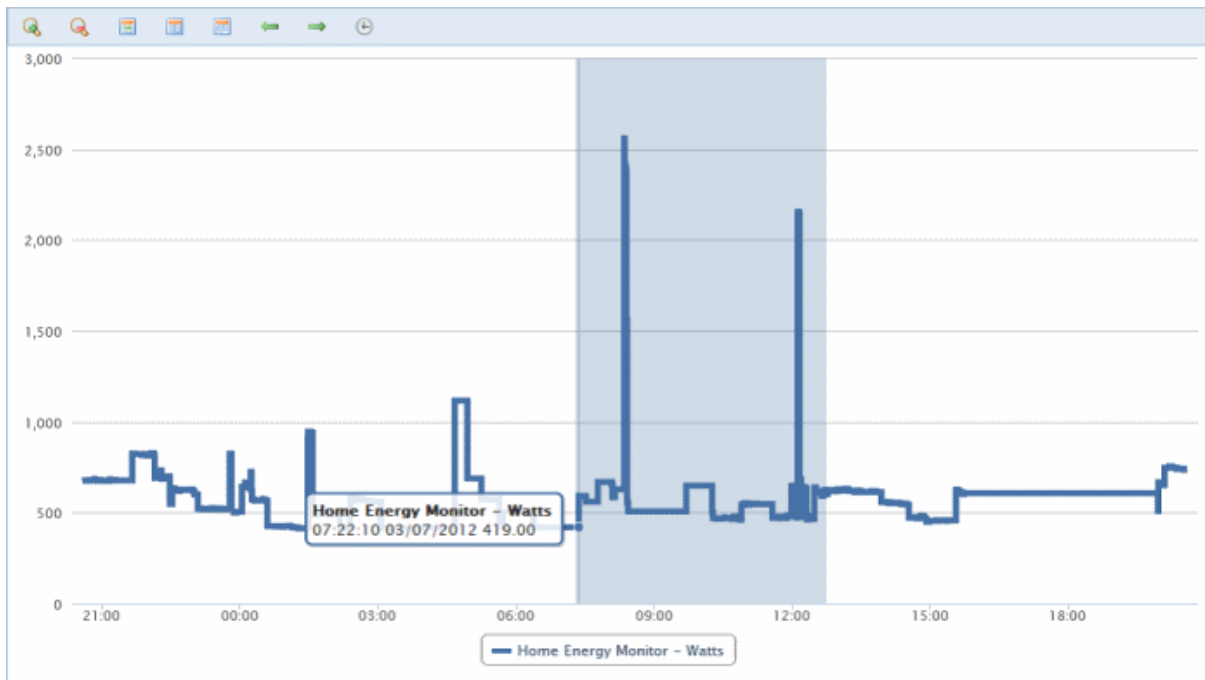


Figure 10. Graph Window Zoom

Clicking Save Graph will save the currently selected charts and show the dialog below. This enables the editing of the Axis number which can be used to force charts to share the same axis and also allows a Quickview reference name to be added which can be used to show a pre-saved graph on a separate web page. This is useful for external apps such as Imperihome. Saved graphs can also be retrieved from the Saved Graphs panel.

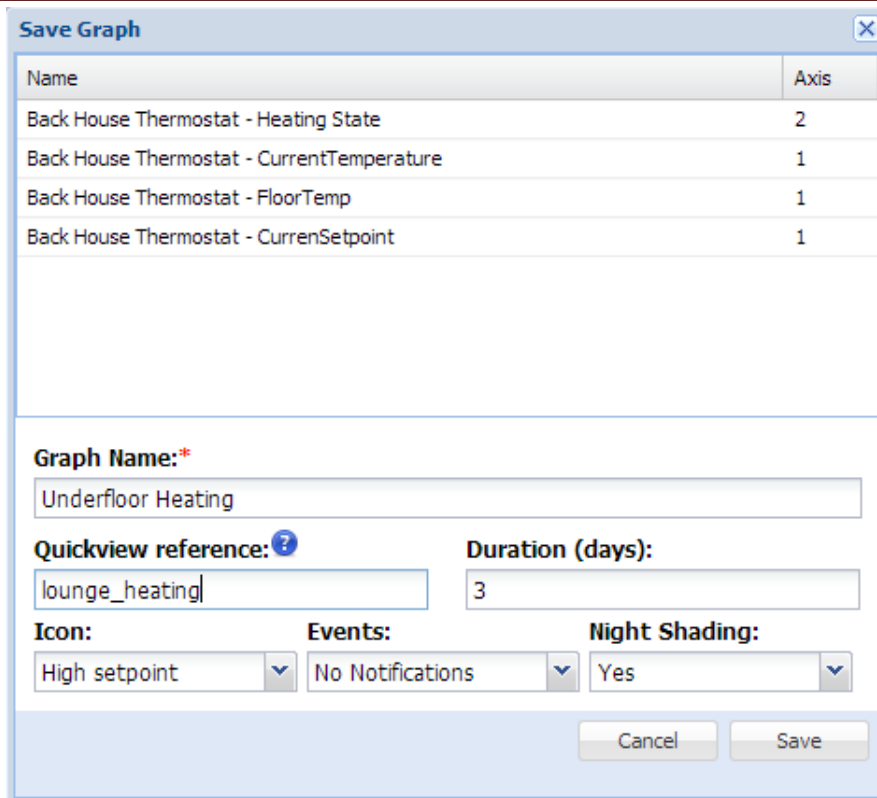


Figure 11. Graph Save / Edit window

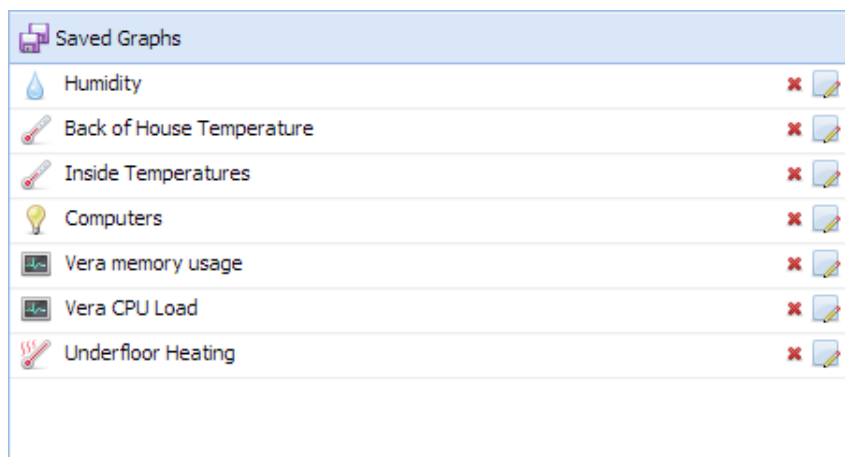


Figure 12. Saved graph selection list.

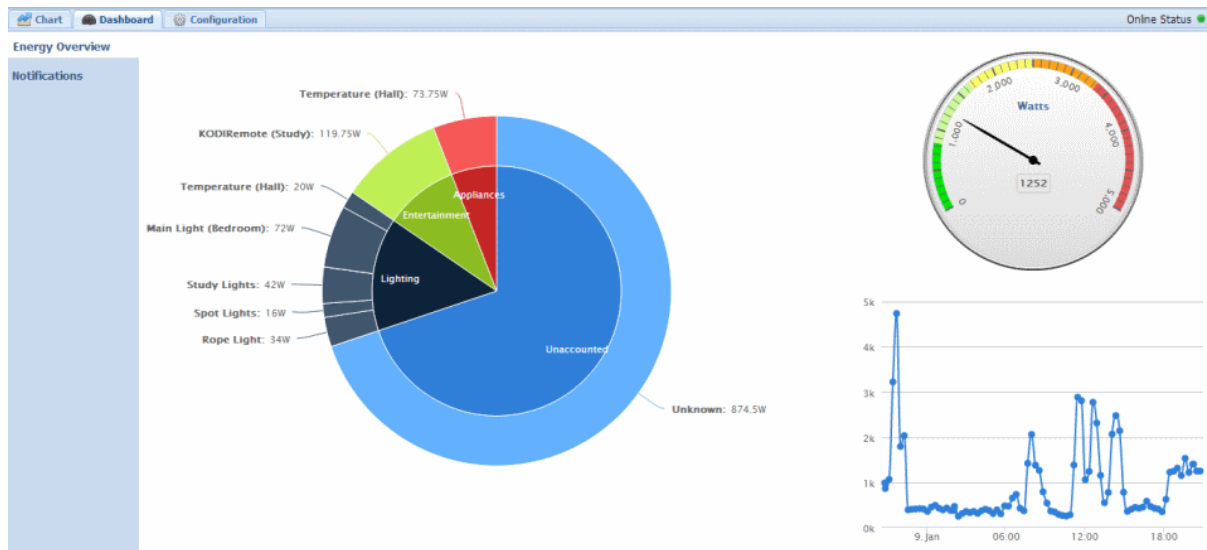
Figure above shows the saved graph selection list. The X icon allows you to delete the saved graph, and the icon allows you to edit the graph data. Note that when editing a graph, if you change the name, *dataMine* will save a new copy of the graph and you will need to delete the one you don't want.

Dashboard Tab

Figure below 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

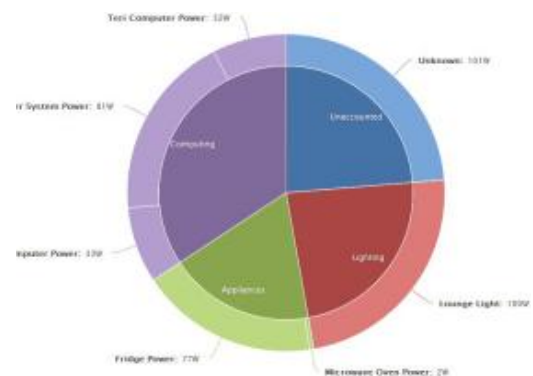
DataMine2 for Vera Overview and User Guide

chart showing the split by device and category. A trend chart shows the whole-house use over the past 24 hours.



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).



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 <http://vera-ip/dm/graph.html?ref=<quickref>&period=<days>&update=<mins>>

e.g. <http://vera-ip/dm/graph.html?ref=heating&period=1>

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.

DataMine2 for Vera Overview and User Guide

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.

dataMine2 also provides an alternative page: graph2.html which uses Google Charts to display the data; this also renders line/area charts as column charts. Graph2.html also uses the saved details of the aggregation type.

e.g. <http://vera-ip/dm/graph2.html?ref=heating>

Advanced: Using the API

There are several API calls available from DataMine2 they take the following format with various request parameters. The below list is not exhaustive but includes the main ones.

http://vera-ip/port_3480/data_request?

Debug Information

Parameter	Example Value	Description
id	Lr_dmCtrl	Mandatory tell the system to return system data
control	debug	Return debug information in 7 sections: 1. Output of: <code>cat /var/log/cmh/LuaUPnP.log grep -i dataMine</code> 2. Output of: <code>blkid</code> 3. Output of: <code>mount</code> 4. Output of: <code>fdisk</code> 5. Values of 4 device variables 6. Json encoded configuration data 7. End marker

Example request and response:

http://vera-ip/port_3480/data_request?id=Lr_dmCtrl&control=debug

```
-1-----
50      01/07/16 23:15:59.994 luup_log:37: dataMine: 2:Finished reading
files - no more data available/dataMine/database/29/raw/2402.txt
<0x2ead7680>
-2-----
/dev/sda1: LABEL="VERA" UUID="6A0D-39FD"
-3-----
rootfs on / type rootfs (rw)
/dev/sda1 on /dataMine type vfat
-4-----
Disk /dev/mtdblock0: 0 MB, 524288 bytes
  Device Boot      Start         End      Blocks   Id System
/dev/sda1   *            1           995     127312+   b Win95 FAT32
-5-----
SetDataDirectory-/dataMine/
SetMountUUID     -6A0D-39FD
SetMountPoint    -
```

DataMine2 for Vera Overview and User Guide

```
SetManualMount -0
-6-----
{"Version":"1.896","dbVersion":2,"Events":{"count":144,"last":1452100637}
-7-----
```

Data Request

Parameter	Example Value	Description
id	Lr_dmData	Mandatory tell the system to return data
start	1451569377	UNIX timestamp for the data start period
stop	1452174177	UNIX timestamp for the data end period
channelX	36	Up to 10 channels can be specified e.g. channel1=36. This is the channel number for the variable to be returned. Channel numbers can be seen on the Configuration tab for the web interface. At least 1 channel must be specified
aggbyX	3	Optional aggregation period for the specified channel. Auto = 0, none = 1, hour = 2, day = 3, week = 4, month = 5
aggtypeX	6	Optional aggregation period for the specified channel. Sample = 0, difference = 1, sum = 2, average = 3, minimum = 4, maximum = 5, time on = 6

Example request and response:

http://vera-ip/port_3480/data_request?id=Lr_dmData&start=1451569377&stop=1452174177&channel1=36&aggby1=3&aggtype1=6

```
{"series":[{"label":"Heating -
Status","Id":36,"data":[[1451569377000,2.18888888888889],[1451606400000,4.44
7222222222],[1451692800000,2.915],[1451779200000,4.05277777777778],[1451865
600000,3.2597222222222],[1451952000000,4.4275],[1452038400000,5.25],[145212
4800000,0.9972222222222]],"pointsRet":8,"pointsTot":180,"min":0.997222222
2222,"max":5.25}], "procTime":0.02999999999999973,"min":1451569377,"max":14521
74177}
```

Variable List

Parameter	Example Value	Description
id	Lr_dmList	Mandatory tell the system to return list of variables

No other parameters are required. Example request and response:

http://vera-ip/port_3480/data_request?id=Lr_dmList

```
[{"Ghost":false,"Service":"urn:upnp-
org:serviceId:Dimming1","AggregateBy":1,"DrowsyError":0,"AggregationType":0
,"LastRec":1452159609,"Units":"0","Id":1,"FilterMaximum":0,"Logging":1,"Typ
e":1,"FilterEnable":0,"DrowsyWarning":0,"Device":3,"FilterMinimum":0,"DataM
ultiplier":1,"LastHistory":1104516000,"DataType":0,"DataOffset":0,"Name":"R
ope Light -
LoadLevelStatus","LastVal":0,"FirstRec":0,"Variable":"LoadLevelStatus","his
toryState":0},{ "Ghost":false,"Service":"urn:micasaverde-
com:serviceId:ZWaveDevice1","LastRec":1452161230,"AggregateBy":1,"DrowsyWar
ning":0,"AggregationType":0,"DrowsyError":0,"LastHistory":1104516000,"First
Rec":0,"FilterMaximum":0,"LastVal":"false","Type":1,"DataMultiplier":1,"Dat
```

DataMine2 for Vera Overview and User Guide

```
aType":0,"Device":36,"FilterMinimum":0,"Alpha":1,"Units":"","Id":59,"DataOffset":0,"Name":"Temperature (Hall) - isNight", "FilterEnable":0,"Logging":1,"Variable":"isNight", "historyState":0, "Lookup":{"true":1,"false":0}}}
```

Sunrise Data

Parameter	Example Value	Description
id	lr_dmCtrl	Mandatory tell the system to return system data
control	sunrise	Return sun set and sun rise data

This request returns a list of timestamps for sunrise and sunset. Example request and response:

http://vera-ip/port_3480/data_request?id=lr_dmCtrl&control=sunrise

```
{"Sunrise":[{"time":1438403320,"night":false}, {"time":1438458639,"night":true}, {"time":1438489811,"night":false}, {"time":1438544938,"night":true}, {"time":1438576302,"night":false}, {"time":1438631237,"night":true}, {"time":1439008765,"night":false}, {"time":1439062702,"night":true}, {"time":1452096657,"night":true}, {"time":1452154075,"night":false}, {"}], "procTime":0, "last":0}
```

Graph List

Parameter	Example Value	Description
id	lr_dmCtrl	Mandatory tell the system to return system data
control	listGraphs	Return graph list data

This request returns a list of graphs that have been saved within the web interface. For each graph, information is provided on the channels and variables logged. Example request and response:

http://vera-ip/port_3480/data_request?id=lr_dmCtrl&control=listGraphs

```
[{"Night":1,"Period":604800,"Channels":[{"Device":34,"Service":"urn:upnp-org:serviceId:TemperatureSensor1","yAxis":2,"Variable":"CurrentTemperature"}, {"Device":22,"Service":"urn:upnp-org:serviceId:TemperatureSensor1","yAxis":2,"Variable":"CurrentTemperature"}, {"Device":16,"Service":"urn:upnp-org:serviceId:TemperatureSensor1","yAxis":2,"Variable":"CurrentTemperature"}, {"Device":16,"Service":"urn:upnp-org:serviceId:SwitchPower1","yAxis":3,"Variable":"Status"}],"Events":0,"Icon":1,"Reference":"temps","Name":"Temperatures"}]
```

Google Charts Data Request

Parameter	Example Value	Description
id	Lr_dmGoogle	Mandatory tell the system to return data
start	1451569377	UNIX timestamp for the data start period
stop	1452174177	UNIX timestamp for the data end period

DataMine2 for Vera Overview and User Guide

channelX	36	Up to 10 channels can be specified e.g. channel1=36. This is the channel number for the variable to be returned. Channel numbers can be seen on the Configuration tab for the web interface. At least 1 channel must be specified
aggbyX	3	Optional aggregation period for the specified channel. Auto = 0, none = 1, hour = 2, day = 3, week = 4, month = 5
aggtypeX	6	Optional aggregation period for the specified channel. Sample = 0, difference = 1, sum = 2, average = 3, minimum = 4, maximum = 5, time on = 6

Use this request to return data in a format that can be consumed by Google Charts. For a code sample using Google Charts see: <http://vera-ip/dm/GoogleChart.html>

Example request and response:

http://vera-ip/port_3480/data_request?&id=lr_dmGoogle&start=1451570819&stop=1452175619&channel1=36&aggby1=3&aggtype1=6

```
{ "rows": [ { "c": [ { "v": "Date (1451606400000) " }, { "v": 4.447222222222222 } ] }, { "c": [ { "v": "Date (1451692800000) " }, { "v": 2.915 } ] }, { "c": [ { "v": "Date (1451779200000) " }, { "v": 4.052777777777778 } ] }, { "c": [ { "v": "Date (1451865600000) " }, { "v": 3.259722222222222 } ] }, { "c": [ { "v": "Date (1451952000000) " }, { "v": 4.4275 } ] }, { "c": [ { "v": "Date (1452038400000) " }, { "v": 5.25 } ] }, { "c": [ { "v": "Date (1452124800000) " }, { "v": 0.997222222222222 } ] } ], "cols": [ { "id": "0", "type": "date", "label": "Time of Day" }, { "id": "36", "type": "number", "label": "Heating - Status" }
```

Setting System Defaults

Parameter	Example Value	Description
id	Lr_dmCtrl	Mandatory tell the system to return system data
control	saveConfig	Save the values below as config
energyWholeHouseMin	0	Optional: Sets the minimum value of the whole house gauge. Cannot be set above 0.
energyWholeHouseMax	6000	Optional: Sets the maximum value of the whole house gauge. Cannot be set less than 0.
generalDrowsyWarning	3600000	Optional: Sets the drowsy warning system default. Values in milliseconds – defaults to 1 hour (3600000)
generalDrowsyError	86400000	Optional: Sets the drowsy error system default. Values in milliseconds – defaults to 1 day (86400000)
graphDefaultPeriod	1	Optional: Sets the default graph period. Values in days – defaults to 1 day. Must be between 1 to 30.

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. Example request and response:

http://vera-ip/port_3480/data_request?&id=lr_dmCtrl&control=saveConfig&energyWholeHouseMax=5000

DataMine2 for Vera Overview and User Guide

```
{"_dc":"1445963387329","energyWholeHouseMax":"5000","graphOptionShadows":true,"graphOptionAnimation":false,"graphOptionWidth":"5","graphOptionMarker":false}
```

Advanced: 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). Files are then split by week 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 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.

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.

Advanced: 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.

DataMine2 for Vera Overview and User Guide

3. Put a line in you 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 +%Y-%m-%d`.tar.gz $destdir/datamine_logs
# fix rights..
chmod 600 $destdir/datamine_`date +%Y-%m-%d`.tar.gz
# remove tempfiles
rm -rf $destdir/datamine_logs
# Remove all backups created 2 months ago
rm -f $destdir/datamine_`olddate +%Y-%m-%d`.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
```

Advanced: List of Files Used in Distribution

http://code.mios.com/trac/mios_datamine_2 is published under the GPL license.

Filename	Destination Name	Destination Path	Compress
D_DataMine2.json	D_DataMine2.json		Y
D_DataMine2.xml	D_DataMine2.xml		Y
I_DataMine2.xml	I_DataMine2.xml		Y
L_DataMine2.lua	L_DataMine2.lua		Y
S_DataMine2.xml	S_DataMine2.xml		Y
dataMine2Web.tar.gz	dataMine2Web.tar.gz	/../../www/dm/	
json-dm2.lua	json-dm2.lua	/../../usr/lib/lua/	
Data_Mine2.png	Data_Mine2.png	/../../www/cmh/skins/default/icons	