DataStream Configuration Guide

DataStream 800 Series – Firmware V1.5.0 up to V1.6.4

SYN-PDS-0004 W

Author: David Norman
Date: 01-May-2014

Contents

Introduction

The DataStream is an Internet connected device for the collection, storage and transmission of meter data for energy management and environmental monitoring.

Powered by a high speed ARM® processor the DataStream has the capacity to acquire, process, record, present and stream data from multiple energy meters, sensors and networked devices. The DataStream has pulse inputs for direct connection to energy meters (Electric, Gas, Water, Heat Meters for example) and analogue inputs for use with the LEM range of AC Current Transformers. The Modbus serial port enables communication with additional meters, Building Management Systems, PLCs and other networkable devices. TCP Modbus devices can also be read using the Ethernet network connection.

Energy consumption is recorded at selectable intervals, typically ½ hourly. This data is stored on the DataStream and automatically sent to a central server, such as the **enVision** energy dashboard software (www.synetica.net/envision) or a spread sheet such as Microsoft® Excel® for further analysis.

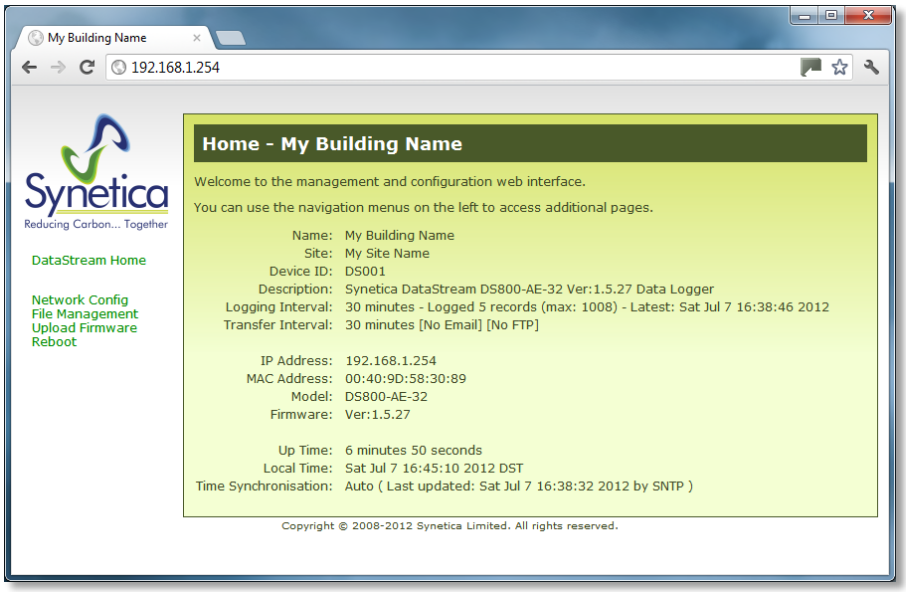
It has several Internet protocols that it uses to deliver logged data. It can send files by email and/or FTP to an FTP Server. It keeps its internal clock synchronised with a time server, or if unavailable, you can set the time from a simple telnet client. Configuration pages and charts can be viewed with any web browser (Microsoft Internet Explorer, Mozilla Firefox etc.). Changes to its configuration can be achieved with an FTP client (FileZilla, Smart FTP etc.) and settings and diagnostics can be performed with a Telnet client (telnet via a command prompt, or HyperTerminal etc.).

The DataStream can log data from digital pulse meters, wired analogue current transformers (CTs), serial RS485/RS232 Modbus RTU devices, Modbus TCP devices and a wide range of EnOcean wireless sensors.

The logged data is pushed either with email or by sending a file to an FTP server. This stops the requirement for a computer to gather the data (manually or automatically) from the DataStream. This data is typically sent in real-time. So a reading that has just been taken can be viewed seconds later. This is faster than a typical data logger that sends data a day after it was measured or 'day plus one'.

The DataStream has two storage areas where it keeps various files. The single configuration file (config.ini) and database (datafile.dat) are kept on non-volatile memory; FLASH memory. The CSV files are stored on RAM which is faster to read from.

The web pages allow you to see at a glance the summary status of the DataStream.



With simple charting showing the data you have logged.

Sample CSV file showing data format:

```
UTCTimeStamp,LocalTimeStamp,Main (kWh),Sockets (kWh),Ground Floor (kWh),UPS (kWh),Temp (DegC),Humidity (%rH)¶
03/07/2009 16:00,03/07/2009 17:00,0.953155,0.299499,0.275164,1.370207,22.08,44.4¶
03/07/2009 16:30,03/07/2009 17:30,0.907481,0.007487,0.278908,1.249659,22.24,44¶
03/07/2009 17:00,03/07/2009 18:00,0.819878,0,0.25083,1.084185,22.08,43.2¶
03/07/2009 17:30,03/07/2009 18:30,0.831858,0,0.207777,1.074452,21.92,42.4¶
03/07/2009 18:00,03/07/2009 19:00,0.814637,0,0.204034,1.042256,21.92,43.2¶
03/07/2009 18:30,03/07/2009 19:30,0.823622,0,0.20029,1.038512,21.92,42¶
03/07/2009 19:00,03/07/2009 20:00,0.810144,0,0.198418,1.025034,21.76,41.6¶
03/07/2009 19:30,03/07/2009 20:30,0.826617,0,0.198418,1.042256,21.76,41.2¶
03/07/2009 20:00,03/07/2009 21:00,0.796667,0,0.198418,1.020542,21.6,41.6¶
```

¶ = CR + LF characters

Here's the same data opened in Microsoft Excel®

A1 fx UTCTimeStamp

	A	B	C	D	E	F	G	H	I
1	UTCTimeStamp	LocalTimeStamp	Main (kWh)	Sockets (kWh)	Ground Floor (kWh)	UPS (kWh)	Temp (DegC)	Humidity (%rH)	
2	03/07/2009 16:00	03/07/2009 17:00	0.953155	0.299499	0.275164	1.370207	22.08	44.4	
3	03/07/2009 16:30	03/07/2009 17:30	0.907481	0.007487	0.278908	1.249659	22.24	44	
4	03/07/2009 17:00	03/07/2009 18:00	0.819878	0	0.25083	1.084185	22.08	43.2	
5	03/07/2009 17:30	03/07/2009 18:30	0.831858	0	0.207777	1.074452	21.92	42.4	
6	03/07/2009 18:00	03/07/2009 19:00	0.814637	0	0.204034	1.042256	21.92	43.2	
7	03/07/2009 18:30	03/07/2009 19:30	0.823622	0	0.20029	1.038512	21.92	42	
8	03/07/2009 19:00	03/07/2009 20:00	0.810144	0	0.198418	1.025034	21.76	41.6	
9	03/07/2009 19:30	03/07/2009 20:30	0.826617	0	0.198418	1.042256	21.76	41.2	
10	03/07/2009 20:00	03/07/2009 21:00	0.796667	0	0.198418	1.020542	21.6	41.6	
11									
12									
13									

What's new?

The manual is written for firmware version 1.5.0 onwards.
Small changes between versions will be noted in the associated section.

- **Modbus TCP Master** feature allows the logging of Modbus devices that support the protocol over TCP
- **Modbus Reference Register** feature enables Analogue CT and Wi-CT to use a voltage from a logged Modbus device, rather than a fixed value. See: **Analogue Current Transformer Inputs** and **EnOcean Inputs**
- **Enable/Disable Web Charts** [1.5.1] Added INI option to Enable/Disable creating CSV files for charts – to help with speed/size issues. See **Storage Guidelines**
- **SR-MDS EnOcean Light Level/Movement Sensor** [V1.5.2] added to EnOcean types
- **Differential Value Logging** [V1.5.3] Extra option in bit-field selector of Modbus devices to specify that the stored value in the CSV file is a differential calculation. Due to the way this works, the first data row in the CSV file will be the value read from the meter, not the differential value. If the user wants, this line can be removed by the DataStream. Use [File] Format = 0 in the INI file
- **Modbus Error Handling in Telnet Client** [V1.5.3] If there is a problem reading a Modbus meter, for example an incorrect ID or serial setting is used, then the DataStream will keep trying to read the device to give it plenty of time to respond. If this happens, then the serial port is in effect blocked until all devices are tried, so you cannot diagnose the problem during this period. To fix this, a new menu has been developed that allows you to see which device(s) are causing the error, and to stop the background task that is trying to read the Modbus devices. Then you can use the various menu options to track down the problem and fix it. Details are in the Modbus section.
- **EnOcean Radio Packet Count Logging** [V1.5.7] Extra option in bit-field selector of EnOcean devices to log the count of received radio telegrams from EnOcean devices.
- **Transfer Interval can be daily or weekly** [V1.5.20] You can now select to transfer the CSV file daily, at a specific time or weekly on a particular day at a specified time.
- **New model available: DS800-128** [V1.5.25] The DataStream can now reference and store up to 128 columns in the CSV file.
- **Modbus RTU Retries are now configurable** [V1.5.25] The default number of retries has been changed to 3. You can alter this from zero to 12.
- **Modbus RTU Timeout is now configurable** [V1.5.38] The default Timeout is 2 seconds. You can alter this from 1 to 20.
- **Wi-LEM Scanner software compatible** [V1.6.0] A Windows application has been developed to allow re-time diagnosis of Wi-LEM networks. This DataStream firmware version is required for the application.

DataStream Models

There are three main models of the DataStream:

- **DS800-32**
- **DS800-64**
- **DS800-128**

The first model has a capacity to log up to 32 columns of data into the CSV file, this doesn't include the two, optional time columns. The second can store 64, and third up to 128 columns of data.

From these main models, there are three options:

- Analogue **(A)** – This model includes extra hardware to enable the analogue input channels, so it can read from the analogue LEM current transformers.
- EnOcean **(E)** – This model includes an EnOcean transceiver so that it can receive EnOcean data from various wireless sensors.
- Modbus TCP Slave **(M)** – This model includes extra software to enable the DataStream to act as a Modbus Slave on the TCP network. This enables PC based software or other devices to interrogate the DataStream's internal registers in real-time. See the Appendix at the back of this guide for the register map.

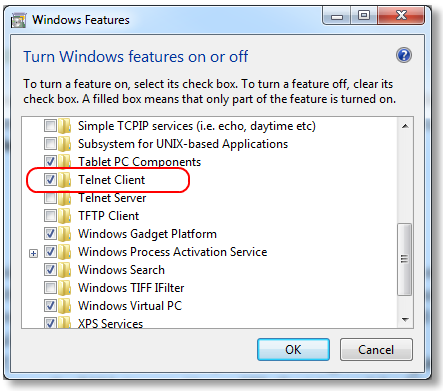
These options are shown in the various model codes, for example:

- **DS800-M-32** – this is the standard DataStream with 32 column capacity plus the Modbus TCP Slave option
- **DS800-A-64** – this is a 64 column DataStream, with analogue CT inputs enabled
- **DS800-AE-32** – a 32 column DataStream with analogue CT inputs, and EnOcean wireless transceiver options

Quick Start

Utilities for your laptop or computer

To help you configure your DataStream, we recommend that you have a Web Browser, Telnet client, FTP client and a text editor. If you are using Windows 7 or 8, the telnet client is not enabled by default. Use your control panel, programs applet and enable the Telnet Client in the **Turn Windows features on or off** option:



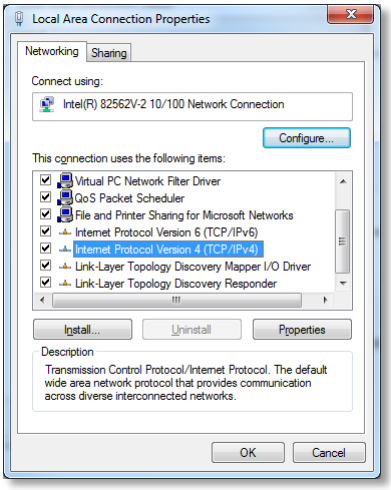
We recommend using FileZilla http://filezilla-project.org/ for your FTP Client and Notepad++ http://notepad-plus-plus.org/ for your text editor. These were used in the screenshots in this guide.

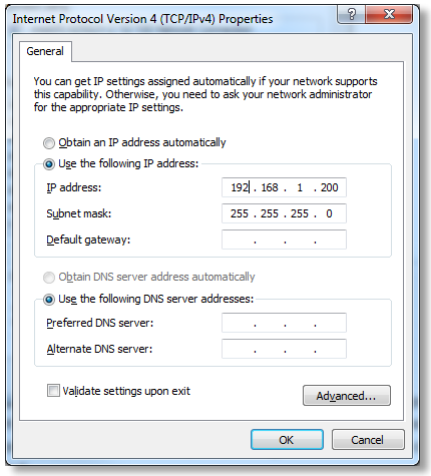
Connect to your DataStream

Plug an Ethernet cable between the DataStream and your computer and then power up DataStream. If your computer has an auto-sensing Ethernet port it will adjust to form a working connection. If not, you will need a cross-over cable.

The DataStream is assigned a default static IP address of **192.168.1.254** with a subnet of **255.255.255.0** during manufacture.

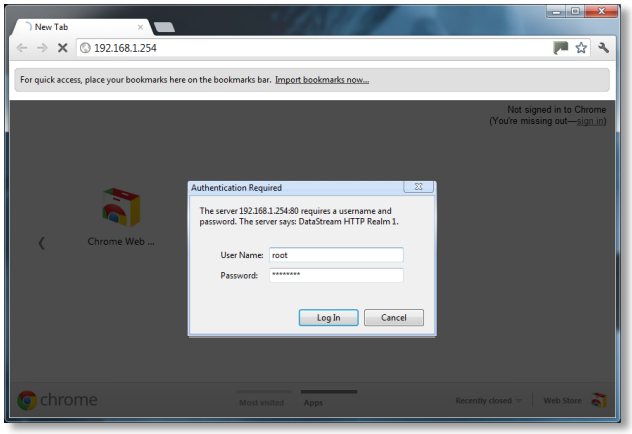
If necessary, change your computer's IP address to a static one, for example: **192.168.1.200** with the same subnet.



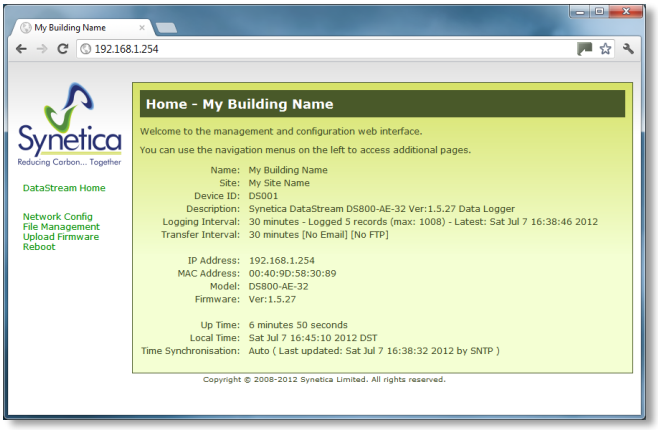


Now, try to contact the DataStream. Use a web browser and enter **192.168.1.254** into the address bar. If your network is correct you will see an authentication dialog in your browser. Use the default username and password:

UID: **root**
PWD: **password**



Logon dialog in Google Chrome web browser.



DataStream home screen.

Configure your DataStream and test your field devices

Now you can configure your DataStream simply by editing the text file **config.ini** located on the non-volatile memory with your FTP Client and text editor.

You can test your field devices connected to your DataStream using your Telnet Client with the various menu options. See the Telnet Client options section, further on in this guide.

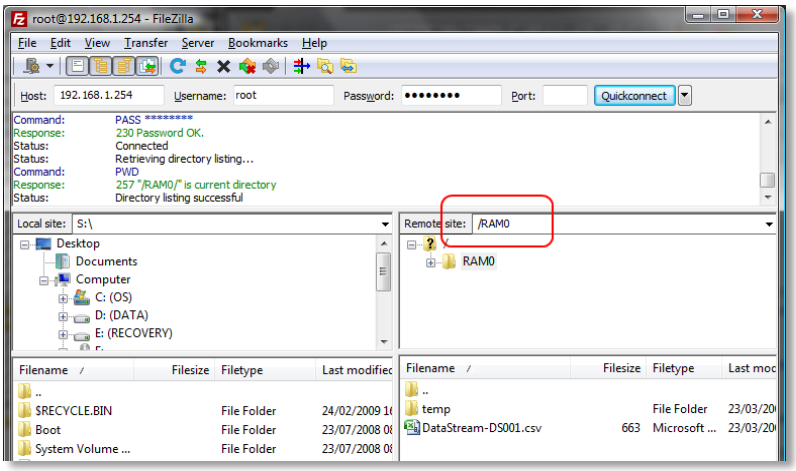
Configuration

Some of the settings in the **config.ini** file are self-explanatory however there are some that require more detail. This document will cover the main configuration settings. An appendix is included giving technical details for the various sections and items of the configuration.

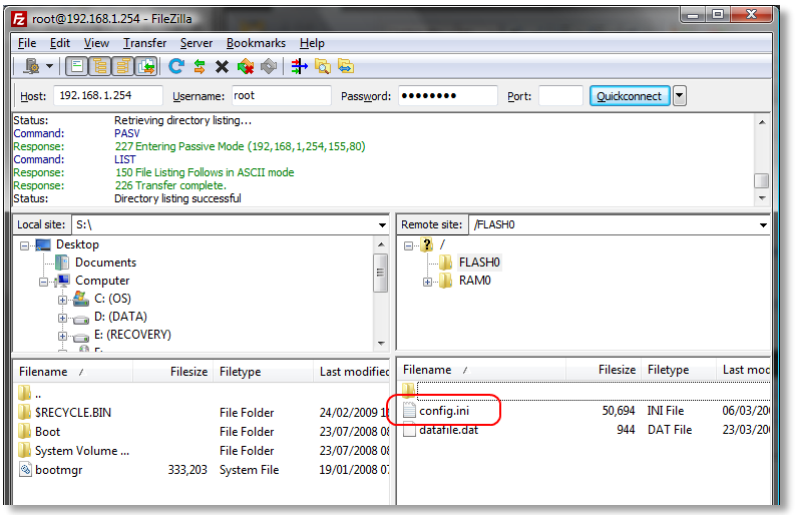
To gain access to the config.ini file, you use an FTP Client, like FileZilla. You should be able to use this information to navigate with other FTP clients.

The DataStream comes pre-configured with a static IP address of 192.168.1.254 and subnet of 255.255.255.0. When you connect it to your network, or directly to your computer, ensure you are running with a similar IP address and subnet, so that you can contact the DataStream IP.

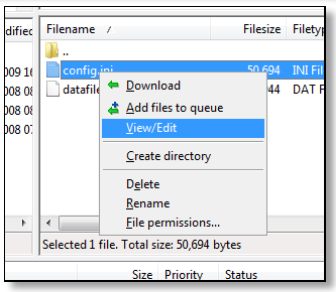
Log on to the DataStream using the default credentials (UID: root PWD: password). By default, the DataStream will open with a path to the RAM area (Shown as '/RAM0'). This allows quick downloading of the CSV data file.



Navigate to the 'FLASH0' FTP folder:
Edit the highlighted area showing '/RAM0', and change it to '/FLASH0' (case sensitive) and hit enter.



You'll see the config.ini file. Right click and select 'View/Edit' – or you could download it and edit the file from there.



Once you have the file open, you can see all the settings.

Once the file has been edited and closed, FileZilla will automatically upload it back to the DataStream (only with the View/Edit option). Once the DataStream detects the file had been changed, it will reset itself and re-boot. To speed up the detection, you should close the FileZilla application to terminate any connections.

If, for example, you have dragged the file to your desktop for editing, you will need to drop it back onto the DataStream file system (ensure you replace the existing config.ini file in the **/FLASH0** folder). The DataStream will reset as before. You will see the Input lights flashing as the DataStream scans the INI file for changes and resets the network etc.

The Configuration File Sections

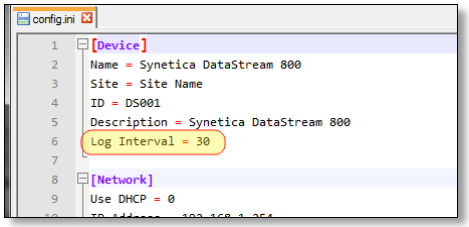
The configuration file sections and order needs to stay the same as stored on the DataStream. However, comment lines may be added to help the user. Up to 5 consecutive comment lines are allowed.

On power up, the DataStream reads the configuration file. If no file exists, a default file is created. If the file is corrupted, or a section or value is not found in the correct order, then this file is renamed to 'config.old.ini' and a new default file is created. This is why it is important that the sections are kept as the original file. All eight Input LEDs will flash, and a file called **config_error.log** is placed on the RAM storage area. See previous **Power Up** section.

When a new firmware is released, this may have extra entries in the configuration file that are incompatible with the existing version. If this is the case, your original file will be renamed to 'config.old.ini'. This gives you the opportunity to copy the required sections and values from your old file to the new format.

The following is a detailed description of important parameters in the configuration file. A summary of every setting can be found in Appendix 1.

[Device] Log Interval

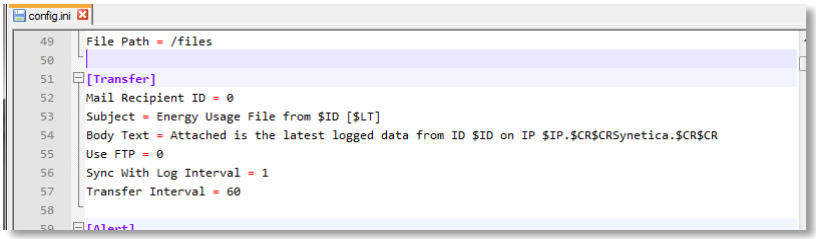


The log interval units are minutes, valid values are:
1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60, 120, 180, 360, 720 and 1440.

If you enter any other value, then a default of **30** will be used.
Normally, you would have the interval that the file was transferred (via email or FTP) the same as the logging interval. You can override this transfer interval in the [Transfer] section.

Scroll down to the [Transfer] section.

You can override the transfer interval by setting the 'Sync With Log Interval' to 0 (zero).
The Transfer Interval can be set to the valid values as used earlier. If an invalid value is used, a value close to a valid one is used.



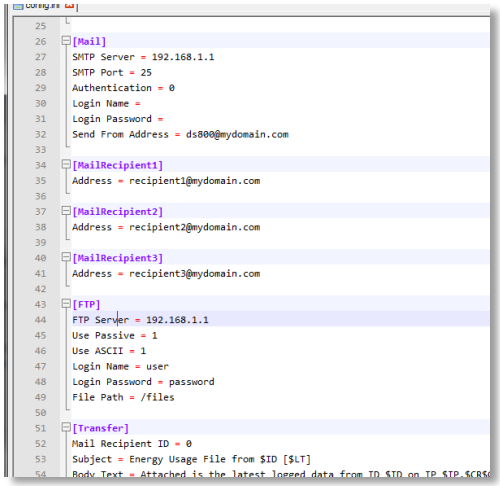
NEW: From Version 1.5.20: You can now select to transfer the CSV file daily (at a specific time) or weekly on a particular day (at a specified time). To do this, you need to set the parameter like the following examples:

- **`Transfer Interval = D 07:30`** This means Daily at 07:30
- **`Transfer Interval = D0 07:30`** This means on Day Zero (Sunday) at 07:30
- **`Transfer Interval = D2 07:30`** This means on Day Two (Tuesday) at 07:30
- **`Transfer Interval = D6 07:30`** This means on Day Six (Saturday) at 07:30

Valid values for the 'D' parameter are 0 through to 6, which correspond for Sunday through to Saturday.

If you are going to use the FTP Client (built in to the DataStream) to send the CSV file to an FTP Server – the setting is:
Use FTP. Set it to 1 to enable, 0 (zero) to disable.

Note: If **Use FTP** is set to 1, then during the start-up procedure the DataStream will send the CSV file via FTP. It won't wait for the first transfer time.
Also, if the Trace email has been enabled (by setting the [Trace] Mail Recipient ID to a non-zero value), an email will be sent that contains the configuration file (and any file saved during a failure).
If both the trace severity trigger is set to 2 and the FTP transfer option is enabled, the configuration file (and failure record file, if it exists) will be sent to the FTP server.



If you want to use the email functionality, you select the user you want to send the file to by changing the 'Mail Recipient ID' from 0 (zero), which disables the email, to 1, 2 or 3.

If you are using network settings that use domain name resolution (often this is available when you select the DHCP option as DNS servers are automatically assigned) you can use fully qualified domain names for the Mail Server and FTP server. If not, stick with IP addresses.

Note: We have had no success using the Google SMTP servers. Instead, try using your corporate SMTP server. If you need an internet based SMTP server, we have successfully used an email account from www.gmx.com

```
[Mail]
SMTP Server = mail.gmx.com
SMTP Port = 25
Authentication = 5
Login Name = username@gmx.com
Login Password = password
Send From Address = username@gmx.com
```

For GMX email to work, ensure you set the **Send From Address** parameter to your login email value.

Authentication:

0 = SMTP_AUTH_NONE
1 = SMTP_AUTH_DIGESTMD5
2 = SMTP_AUTH_CRAMMD5
3 = SMTP_AUTH_LOGIN
4 = SMTP_AUTH_PLAIN
5 = SMTP_AUTH_ANY

Option [5] Checks the server 'ehlo' response for the server supported authentication in the following order:

Digest-MD5
CRAM-MD5
LOGIN
PLAIN

The FTP client will first attempt to connect with either Passive or Active depending on the setting. (Use Passive = 1 for Passive, 0 for Active) If the connection fails it will swap modes and try again. It can use either ASCII (Use ASCII = 1) or Binary (Use ASCII = 0) transfers. ASCII has been used extensively and is found to be reliable.

In the text that is used in sending the emails there are some formatting codes that use substitution for values in the DataStream. These are used to substitute characters to replace a keyword with (often) dynamic information, like time and date. An appendix covers the substitution codes.

Notes:

The [Config] setting is used to email the **config.ini** file as an attachment to an administrator when it is changed. This is intended to be used where several people change the settings, and/or for storing the latest settings file with ease. The [Trace] setting is used to email the **config.ini** and any diagnostic files to an administrator. These can be forwarded to the developers of the DataStream to respond to any firmware problems.

A useful shortcut in Notepad++ is [Alt] + 1. This collapses all the sections and makes it easier to navigate the file.

Testing the Email/FTP file transfer settings - the Telnet Client Menus

Testing the FTP and/or Email settings has been made easy by using the Telnet Client menus.

Email Menu and FTP Menu allows you to change all the INI settings from menus, and test your settings. Your changes are automatically saved in the INI file.

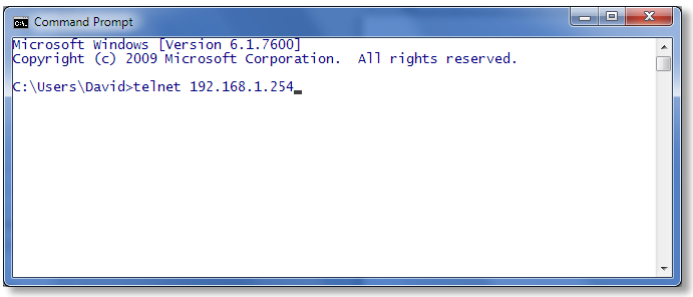
This example uses Microsoft Windows but any telnet client will work.

From a command prompt window type

`telnet 192.168.1.254`

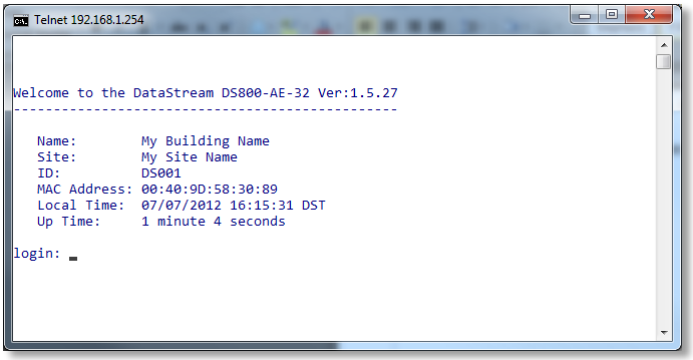
and press enter.

Substitute your DataStream's IP address if different.



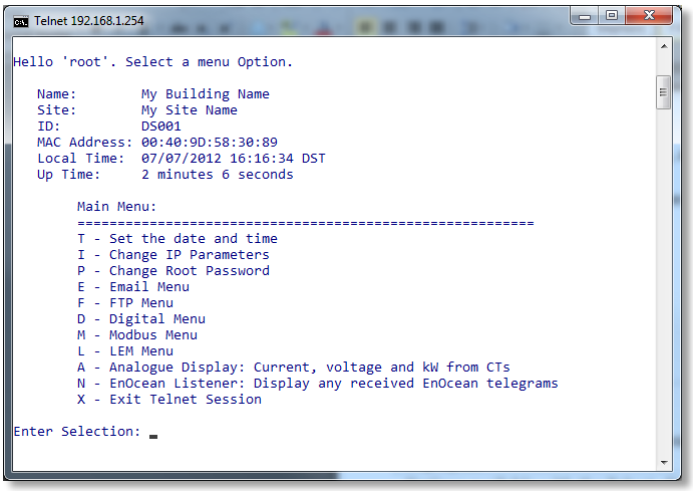
Log in using **root** as the login name, and **password** as the password.

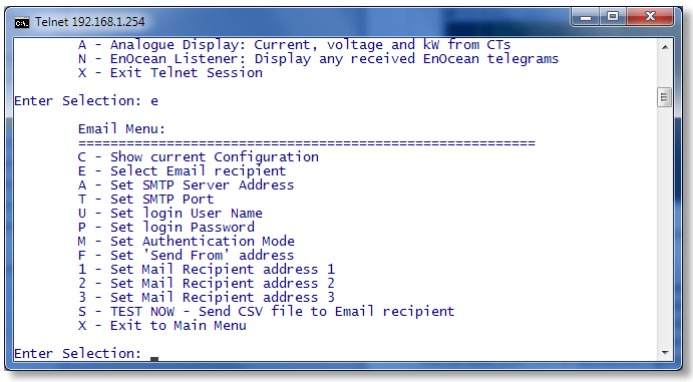
Use a different password if you have changed it.

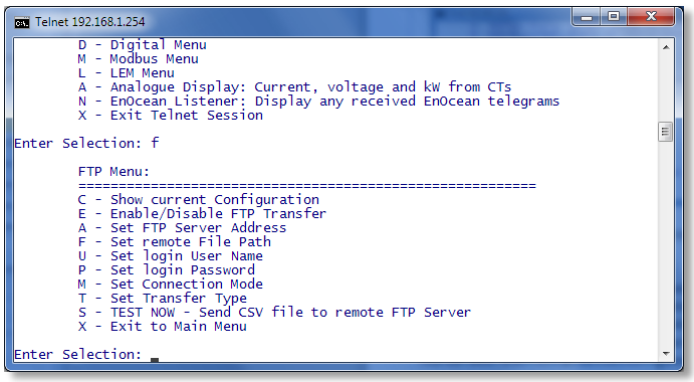


And then select a menu from the available options displayed.

To select the email menu, type **e** and press enter.







Simply follow the on-screen menus.

Logging data into the CSV file

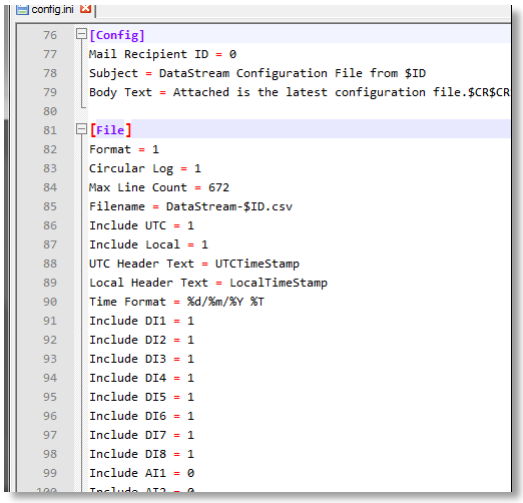
The DataStream creates a small, efficient data file on the FLASH storage area. Using this, it creates the main CSV file and other data files for the charts in the RAM.

Note: Some configuration settings will cause this file to be deleted and rebuilt. This is because its structure is based on the columns you want to log – this is to keep the file as small as possible. The settings that will cause this to happen are:

- **Logging Interval** in the [Device] section
- Any **Include XX** item values in the [File] section
- **Max Line Count** in the [File] section
- **Type** value in the [EnOceanTemplateXX] section.

Ensure you backup any data you want to keep before you save the new configuration.

Firstly we'll look at the [File] section.



Format = 1	The default value for this is '1'. When using the Differential Value feature when logging Modbus values, you can select if you want to remove the first row of data in the CSV file. To do this, set this value to '0' (zero). You can read more on the differential value feature in the Modbus section.
Circular Log = 1	Set this to 1 so that the CSV file removes the older rows (from the top of the CSV file) when the 'Max Line Count' is reached. If you select the 'Circular Log = 0' then the CSV file grows to the 'Max Line Count' then stops logging.
Filename = $ID.csv	This uses the substitution keywords to put the ID value from the first section. You can use any name or extension you want here and also you can use the static substitution keys, as shown in the example above. Do not use any space characters in the filename.
Include UTC = 1 Include Local = 1	If you set these values to 0 (zero), then the selected time column won't be included in the CSV file. You can also enter your own header text for these. The format of the time can also be set using the 'Time Format' string. There is an appendix to detail the time format codes.

The 'Include' values here are slightly more complex. They also depend on the type of input they refer to. Please see how these are used in the following sections.

Column Header Text

The Column headers are normally made up from the Name of the meter or sensor being logged, with the Units text in brackets appended. For example, if you have a meter with the name of Main Electric and units of kWh, then the column header will be:

```
Main Electric (kWh)
```

From version 1.5.16, the header text can be formatted slightly differently. If you set the Units parameter to blank, then the DataStream will remove the brackets and the text will be:

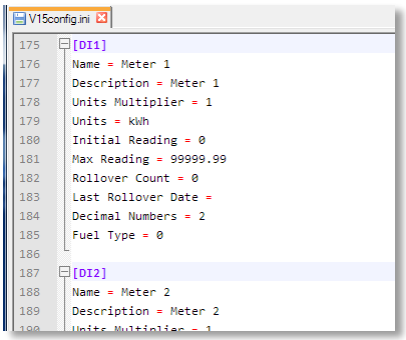
```
Main Electric
```

Digital Pulse Inputs

Include DI1 = 1

The value used here is a bit mask.
The number is 5 bit (binary) representation that corresponds to:

[Lowest Bit]	Consumption in Units per Log Interval [binary = xxx0 0001]
	Raw Pulse Count per Log Interval (and Total counts, that's 2 columns) [xxx0 0010]
	Reading in Units (the Meter reading) - uses 'Decimal Numbers' as the precision [xxx0 0100]
	kWh reading (required before CO_2 calculation) [xxx0 1000]
[Highest Bit]	Carbon Dioxide used in gram/kWh per Log Interval [xxx1 0000]



So, to include a column showing the Consumption, set the value to '**1**'.
If you want Consumption and Raw Pulse counts, set it to '**3**'. [xxx0 0011]
If you only want the Carbon Dioxide, set it to '**16**'. [xxx1 0000]
Therefore, up to 6 columns per input can be logged.

Setting it to zero, means no columns for that input will be included in the CSV file.

Note: 'Consumption in Units per Log Interval'. This may trip you up. If you have a logging interval of 60 minutes, and your units are kWh, the values here are kWh. OK, now if you change the interval to 30, you will see the values logged drop by half, this is because you are logging kWh over a half hour period.

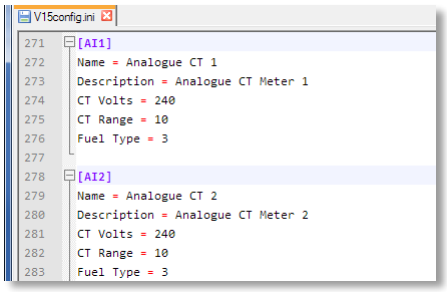
The settings for each input have their own section in the INI file. Logically, these are named [DI1] for Digital Input 1. There are 8 sections.

Name	This text value is used for the Heading in the CSV file (31 characters maximum)
Description	Is not used in the CSV file and is for your reference only (63 characters)
Units Multiplier	This is a floating point number used to multiply against the number of pulses to get the counted value into the Unit value. For example, if you have a pulse meter that counts 1600 pulses per kWh, set this to '0.000625' (1/1600).
Units	This text value is used in creating the Header value in the CSV file (15 characters)
Initial Reading	If you want the DataStream to include the 'Reading in Units' column, then it uses this to calculate what the reading should be based on this initial reading plus the pulse counts so far. 'Reading in Units' means the meter reading, scaled to the Units value.
Max Reading	The DataStream uses the number of significant digits in this value to roll-over the readings. When a roll over occurs, it updates the Rollover count and Last rollover dates in the INI file (for your information only). For example: Let's assume there are 100 pulses per kWh and the Max Reading setting is **99999.99**. The count will go from 00000.00 to 99999.99 then rollover back to zero. Note: The total count is stored in an unsigned 32 bit integer, so in the previous example, if the Max Reading setting is **99999999.99**, then the count will reach **42929672.95** before rolling back to zero.
Decimal Numbers	This is used in formatting the decimal numbers in the 'Consumption' and 'Reading in Units' column when they are written to the CSV file
Fuel Type	This is used to link the metered fuel to a fuel type look-up, so that CO_2 calculations can be made. There are some sample 'Fuel Type' settings at the bottom of the config.ini file. Up to 20 of these can be defined.

Note: Each column used is counted against the total number of columns the DataStream is licensed for. Thus, a DataStream with a 32 column limit will stop at 32 columns (not including the time columns). The 32 isn't used as an input channel counter, so the 8 Pulse Inputs can have a theoretical 48 columns logged.

Analogue Current Transformer Inputs

Include AI1 = 1



The value used here is a bit mask.
The number is 2 bit (binary) representation that corresponds to:

[Lowest Bit]	Consumption in Units per Log Interval
[Highest Bit]	Carbon Dioxide used in gram/kWh per Log Interval

So, to include a column showing the Consumption, set the value to '**1**'.
If you only want the Carbon Dioxide, set it to '**2**'.
If you want Consumption **and** Carbon Dioxide, set it to '**3**'.

Even though there are 8 physical inputs for the Analogue inputs, the configuration allows for 10. This is because 9 and 10 are considered 'virtual meters'. These are '3-Phase' meters that use the physical inputs to totalise the power. Virtual meter 9 uses physical inputs 1, 2 and 3. Virtual meter 10 uses physical inputs 4, 5 and 6.

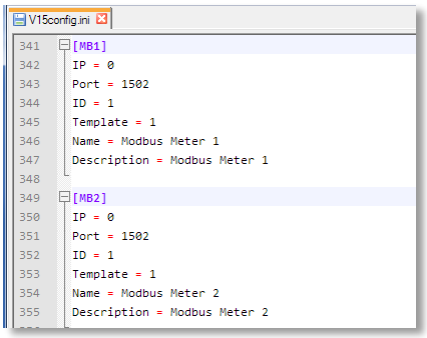
Note: If you select the virtual meters, you must also select the corresponding physical inputs.
The settings for virtual meters don't use the **CT-Volts** or **CT-Range** parameters. Instead, the calculation is based on the average of the settings of the physical inputs.

The DataStream uses an 8 channel, 12 bit A/D convertor, sampling the CTs at 5ms intervals. This measures in instantaneous kW value. The values are integrated over the log interval period (say, 30 minutes) where the kWh value is calculated.

The settings for each input have their own section in the INI file. Logically, these are named [AI1] for Analogue Input 1. There are 10 sections.

Name	This text value is used for the Heading in the CSV file (31 characters maximum)
Description	Is not used in the CSV file and is for your reference only (63 characters)
CT Volts	This integer is used to calculate the power value. **New feature from version 1.5.0:** **Modbus Reference Register** You can use a Modbus reference instead of a fixed value. If you are using a Modbus meter that measures voltage, use its reference number here to calculate the power using the voltage measured with the Modbus Meter. For example, if you are measuring voltage on Modbus input 1 the INI line would look like this: `CT Volts = MB1`
CT Range	This integer (whole number) is used to specify the amperage range of the Current Transformer (CT) connected to the input. These values are: 5, 10, 20, 50, 100, 150, 500, 800, 1000, 1500 and 2000. It represents the current rating of the CT in Amps.
Fuel Type	This is used to link the metered fuel to a fuel type look-up, so that CO_2 calculations can be made. There are some sample 'Fuel Type' settings at the bottom of the config.ini file. Up to 20 of these can be defined.

Include MB1 = 1



The value used here is a bit mask.
The number is 5 bit (binary) representation that corresponds to:

[Lowest Bit]	Converted value in Units
	Raw reading (in the raw data type)
	kWh reading (required before CO2 calculation)
	Carbon Dioxide used in gram/kWh per Log Interval
[Highest Bit]	Differential value (See Below) *New in V1.5.3*

To include a column showing the Converted value, set the value to 1.
If you want the Converted value **and** the Differential value, set it to 17. [xxx1 0001 in binary]

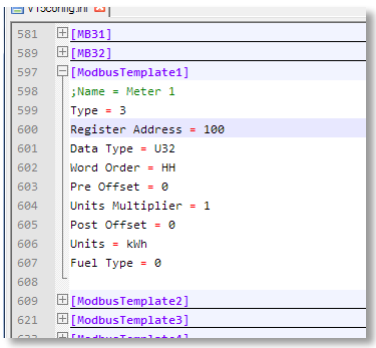
The settings for each register value you want to store have their own section in the INI file. There are 32 sections for DataStream models with 32columns (e.g. DS800-32); 64 sections for DataStream models with a 64 columns, and 128 for the DataStream models with 128 columns.

The Modbus settings also use another concept to make configuration simpler. That is **templates**. Each setting points to a template that stores repetitive data, like Register Address, Data Type and scaling information. Using a template means you can create a single template to monitor power values from a specific type of meter. Then, you can use multiple meters connected to the DataStream, and simply configure each to point to a single template.

IP	**New for Version 1.5.0: Modbus TCP Master Reads** This value is used to determine if the register you want to read is from a TCP or Serial RTU device. Use zero '0' for a Serial Device, or an IP address for a TCP device. From Version 1.5.21 you can use this option to select which serial port you want to use for each device. Set this to **-1** or **-2** to correspond with either Port 1 – the RS485 or Port 2 – the RS232 Bus. Both ports will use the same serial configuration settings.
Port	When reading a TCP register from a TCP Address, you may want to alter the Port value to correspond to the device's own port setting.
ID	This integer value is the Modbus device ID. It can range from 1 to 247.
Template	This is an integer in the range of 1 to 16 (or 1 to 32 for models with 64 columns, or 1 to 64 for models with 128 columns). It points to the template used by this device to gather data.
Name	This text value is used for the heading in the CSV file (31 characters maximum)
Description	Is not used in the CSV file and is for your reference only (63 characters)

There are 16 (or 32, or 64) templates that can be used in the configuration file. These are labelled as 'ModbusTemplate1' … 'ModbusTemplate16'. The number of the template is used as a reference to the device settings earlier in the file.

Note: There is a '`;Name`' setting that is marked as a comment with a preceding semicolon. This is for your own use to help describe the device types.



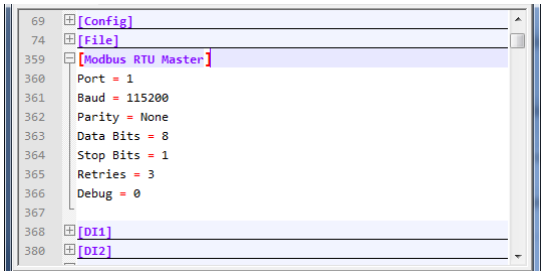
Type	This integer specifies the register type read from the device. Type 3 and 4 are the only value currently supported. This is a 16 bit register. Type **3 are Holding Registers** and **4 are Input Registers**. This matches the Modbus protocol.
Register Address	This integer specifies the internal register for reading.

Data Type	This specifies how to read and interpret the data. Valid values are: S16, U16, S32, U32, SH8, SL8, UH8, UL8, F32 and Bit0…Bit15. These represent, signed/unsigned 16 bit values, signed/unsigned 32 bit values, signed/unsigned high/low byte, 32 bit floating point and bit values taken from one or two consecutive 16 bit holding registers. The bits are read and saved in the CSV file as 0 or 1.
Word Order	This double character is used to define how 16 and 32bit values are interpreted. Use the following: • HH – High byte first, High word first • HL – High byte first, Low word first • LH – Low byte first, High word first • LL – Low byte first , Low word first The most common usage is HH, but we have managed for each combination. **Note:** We strongly suggest using the telnet diagnostic menu for reading values of new or unfamiliar meters and registers. The online diagnostics can help find the required settings you need for the INI file.

To perform some simple value conditioning, the following three values are used.

Pre Offset	This floating point number is added to the read value (negative numbers will be subtracted)
Units Multiplier	This floating point number is used next to multiply the value (previously added to by pre-offset)
Post Offset	Finally, this floating point number is used to add to the result of the value above.
Units	This text value is used in creating the header value in the CSV file (15 characters)
Fuel Type	This is used to link the metered fuel to a fuel type look-up, so that CO_2 calculations can be made.

The 'Modbus RTU Master' section is used to define the communications parameters when reading from serial connected devices



These next few settings specify the communication parameters.

Port	This value can be 0, 1 or 2. A setting of '0' (zero) specifies that Modbus is off. Even if devices and templates have been defined in the configuration file, they will not be read. Set this value to 1 to enable the RS485 port on the DataStream. Set this to 2 to enable the RJ45 BUS port on the DataStream. This is RS232.
Baud	This integer specifies the baud rate at which the DataStream communicates. Valid values are: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200
Parity	Valid Parity values are: **None**, **Even** and **Odd**.
Data Bits	This integer value has valid values of: 5, 6, 7 or 8.
Stop Bits	This integer value can be either 1 or 2.
Retries	*New from V1.5.25:* Default is 3. This can be set from 0 to 12, and determines the number of retries for each device.
Debug	This value is useful when commissioning a device. If set to 1, more text is displayed on the console port (9 pin RS232 serial port) and during telnet diagnostic sessions. A setting of 2 will also show the UART parameters. This is for advanced use.

Modbus Reference Register

You can use a logged voltage value as the voltage reference when using EnOcean Wi-CT or Analogue CT. To use this feature, you need to log a suitable voltage in the CSV file using a Modbus input. Then, when setting the CT Volts in the INI file for either Analogue or EnOcean inputs, you can use the input value instead of a fixed value.

For example; if you log the mains voltage using a Modbus meter on input 1 [MB1], you can use this in the CT Volts section like this:

```
[AI1]
Name = CT 1
Description = CT Meter 1
CT Volts = MB1
CT Range = 100
Fuel Type = 3
```

Differential Value Logging

You can now use this feature when logging cumulative meter values and you want to store the interval usage from this meter. To enable this option use the **Include MB** setting of the INI file. If you wanted to store this value in the CSV file, then set **Include MB1 = 16**

For example; here's a section from a CSV file when logging cumulative data from a standard Modbus meter.

Local Time Stamp	Modbus Meter (kWh)
25/09/2010 11:00	92609.47
25/09/2010 11:30	92627.76
25/09/2010 12:00	92646.21
25/09/2010 12:30	92664.68
25/09/2010 13:00	92681.00
25/09/2010 13:30	92683.57
25/09/2010 14:00	92686.16
25/09/2010 14:30	92688.73
25/09/2010 15:00	92691.23

What you want is the interval data. So, you take the current reading and subtract the previous reading:

Local Time Stamp	Modbus Meter (kWh)	*Interval (kWh)*
25/09/2010 11:00	**92609.47**	***92609.47***
25/09/2010 11:30	92627.76	*18.29*
25/09/2010 12:00	92646.21	*18.45*
25/09/2010 12:30	92664.68	*18.47*
25/09/2010 13:00	92681.00	*16.32*
25/09/2010 13:30	92683.57	*2.57*
25/09/2010 14:00	92686.16	*2.59*
25/09/2010 14:30	92688.73	*2.57*
25/09/2010 15:00	92691.23	*2.50*

Note that the value in the first data row doesn't have a previous value to subtract, so it uses the meter reading.

If you want to remove this first data row from your CSV file, then you can. Use **Format = 0** in the [File] section of the INI file. You can thing of this as 'Include First data Row'. Set to zero therefore doesn't include it.

When using the differential option, then the charts displayed on the DataStream web pages use this value even if both the reading and the differential values are logged.

Modbus In-Error Menu - Modbus Error Handling in Telnet Client

If there is a problem reading a Modbus meter, for example an incorrect ID or serial setting is used, then the DataStream will try to read the device (depending on number of retries and timeouts) to give it plenty of time to respond. If this happens, then the serial port is in effect blocked until all devices are tried, so you cannot diagnose the problem during this period.

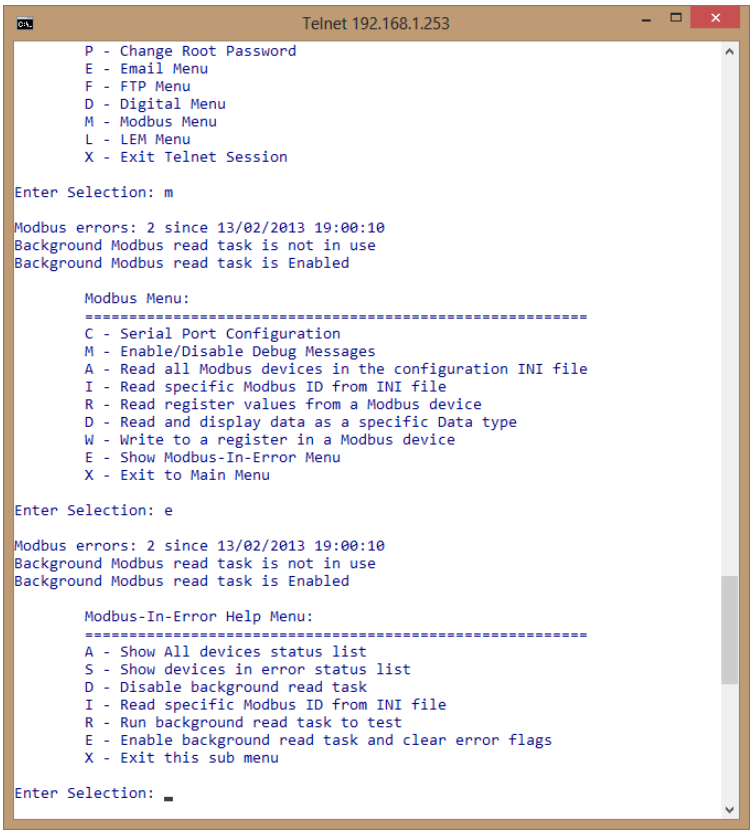
To help you with this, a new Telnet menu has been developed that allows you to see which device(s) are causing the error, and to stop the background task that is trying to read the Modbus devices. Then you can use the various menu options to track down the problem and fix it.

If there are errors, then when you select the Modbus menu, the DataStream will display three summary lines of information above the menu.

To see the errors, select [E] for the 'In-Error' menu.

It will display the three summary lines of information above the menu, as before.

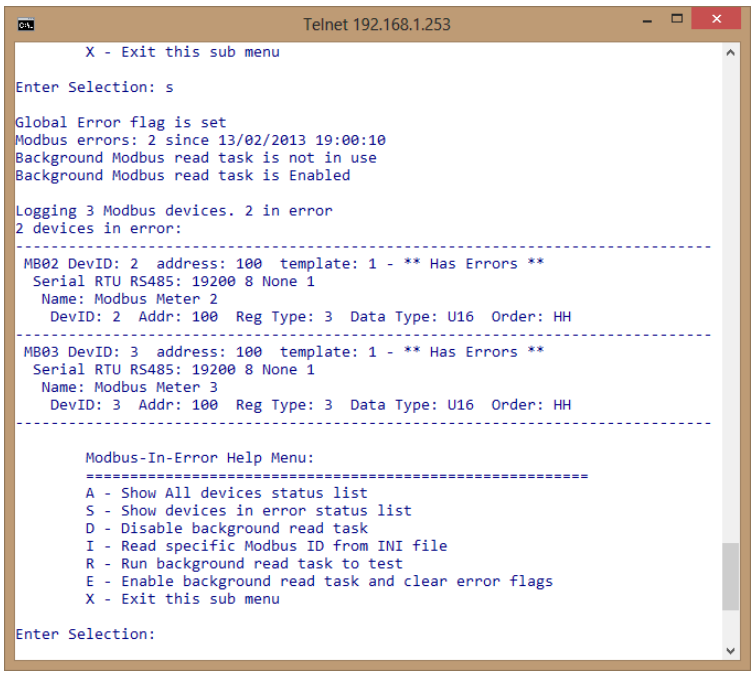
Use the menu options, normally from top to bottom to pinpoint the device(s) with errors.



The first thing you would normally do is to select [S] to show more details.

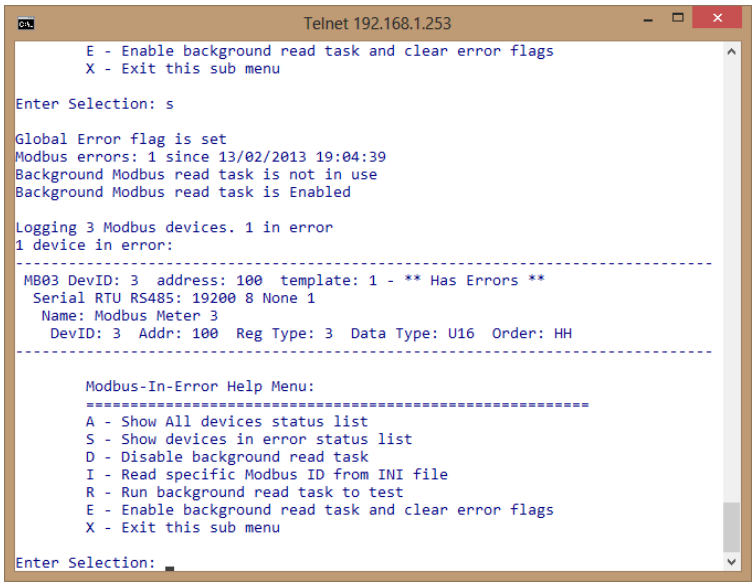
In this example all of the devices are in error. This is often the case if the wrong communication parameters have been set. For example, all the devices are configured at one baud rate, and the DataStream is configured at a different rate. You need to check the settings on the devices, and make sure they are all the same and match the DataStream.

If there are still problems, you can often swap over the communications wires A and B at the DataStream.

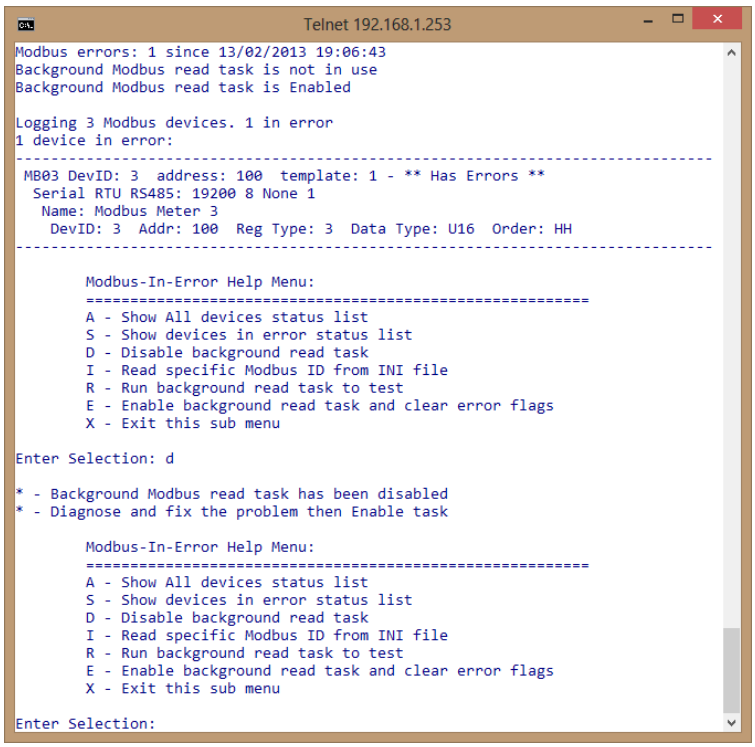


This example shows a single device is in error with others working OK.

This often means the communication parameters are incorrect on this device. Also check the polarity of the A-B connections when using RS485.



The next thing you would do is select [D] to disable the normal logging feature. This will stop the background task from monopolising the COM port and so stopping you from testing the Modbus network.



Now you can use the various menus to track down the problem.

The most useful one is:

```
I – Read Specific Modbus ID from INI file
```

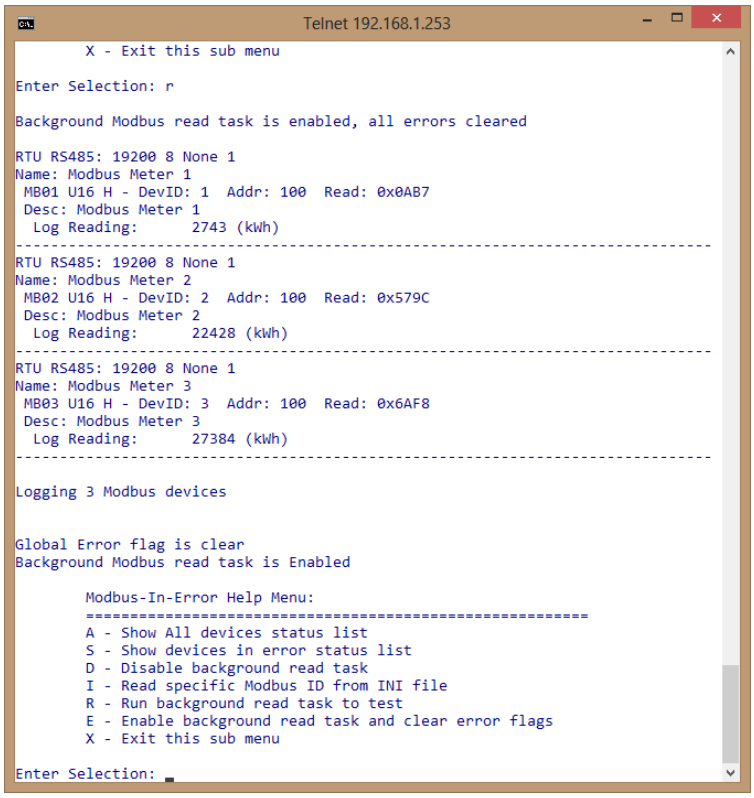
This will enable you to test just one device at a time.

Once you have everything working, you can test every device by selecting:

```
R – Run background read task to test
```

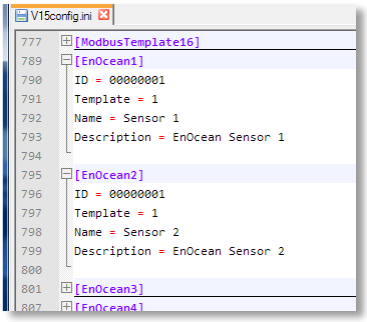
This will read all the devices as listed in the INI file and re-enable the regular running of the background task (at the logging interval).

You can also use the various menus in the normal Modbus menu to change the communications parameters (baud, stop bits etc.) and then re-test.



EnOcean Inputs

Include EnOcean1 = 1



The value used here is a bit mask.
The number is 4 bit (binary) representation that corresponds to:

[Lowest 0x01] Device specific value reading – see below.
[0x02] Carbon Dioxide used in gram/kWh per Log Interval - *Only for Wi-CT*
[Highest 0x04] Packet Count per Interval *New for firmware version1.5.7*

Packet Count per Interval is the count of radio packet telegrams received by the DataStream during the logging interval. This could be used to help re-position sensors when poor reception causes a lower than expected number of received telegrams.

There are 7 supported devices, others will be added as the development of the DataStream continues.
The devices are:

Synetica Wi-CT ..... Wireless current meter device with 3 CT channels
SR04 ..... Indoor temperature sensor with optional set point and fan speed control
SR04rH ..... Temperature/humidity sensor
RTF ..... Room temperature sensor with scaled set point
SR65 ..... Outside temperature sensors
SR-MDS ..... Indoor light level and movement sensor *New: for firmware version1.5.3*
SR07 ..... Indoor temperature sensor *New: for firmware version1.5.23*
Various ..... Light Level Sensors

Some of these devices offer multiple readings that are read by the DataStream and may be logged. These values are defined by a type that is used in the configuration settings to determine what value you want to log. The text for the units needs to be specified in the template.

These types are: (Value range and units stored in the CSV file are shown in square brackets)

0 ..... SR04P and RTF Temperature [0 → 40] Degrees C (0 → 255 reading)
1 ..... Wi-CT Channel 1 – RED marked cable [0 → Max] kWh
2 ..... Wi-CT Channel 2 – GREEN marked cable [0 → Max] kWh
3 ..... Wi-CT Channel 3 – YELLOW marked cable [0 → Max] kWh
4 ..... Three Phase Wi-CT Totals (for Channels 1+2+3) [0 → Max] kWh
5 ..... SR65 Outside Sensor [-20 → 60] Degrees C
6 ..... SR04rH Temperature [0 → 40] Degrees C
7 ..... SR04rH Humidity [0 → 100] %rH
8 ..... Light Sensor [600 → 60000] Lux
9 ..... Light Sensor [300 → 30000] Lux
10 ..... Light Sensor Supply Voltage [0 → 5.1] Volts
11 ..... SR04P and RTF Set point value [0 → 255] no units. Mid-point at 128
12 ..... SR65 VFG and AKF Temperature [10 → 90] Degrees C
13 ..... SR04PST Fan speed stage value [-1 (Auto), 0, 1, 2, 3]
14 ..... Raw data value, data byte 0 [0 → 255]
15 ..... Raw data value, data byte 1 [0 → 255]
16 ..... Raw data value, data byte 2 [0 → 255]
17 ..... Raw data value, data byte 3 [0 → 255]

New for firmware version 1.5.3

18 ..... SR-MDS Light Level [0 → 512] Lux
19 ..... SR-MDS Movement [0 / 1] 0 = No Movement / 1 = Movement
20 ..... SR-MDS Charging voltage [0 → 5.1] Volts

New for firmware version 1.5.23 (September 2011)

21 ..... SR07 and similar Temperature [0 → 40] Degrees C (255 → 0 reading)

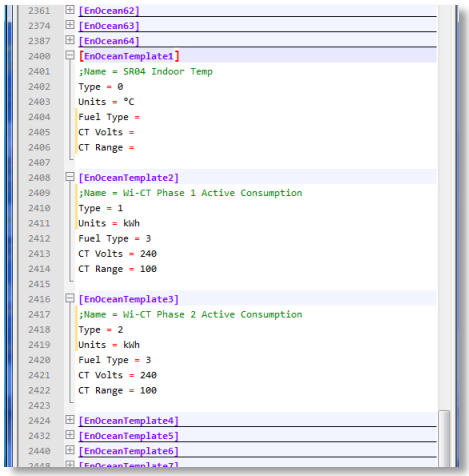
The settings for each device value you want to log have their own section in the INI file.

The EnOcean settings also use templates, where the use of is described in the Modbus section.

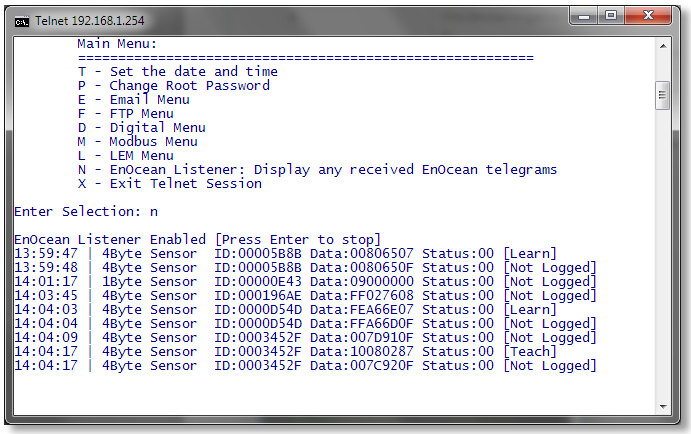
ID	This hexadecimal value is the EnOcean device ID. It is a 4 byte value that ranges from 00000000 to FFFFFFFF.
Template	This is an integer in the range of 1 to 'Maximum Templates' (16/32/64 depending on DataStream Model). It points to the template used by this device to gather data.
Name	This text value is used for the heading in the CSV file (31 characters maximum)
Description	Is not used in the CSV file and is for your reference only (63 characters)

There are 16 templates that can be used in the configuration file. These are labelled as 'EnOceanTemplate1' … 'EnOceanTemplate16'. The number of the template is used as a reference to the device settings earlier in the file.

Note: There is a '`;Name`' setting that is marked as a comment with a preceding semicolon. This is for your own use to help describe the device types.



To discover an unknown device ID, use the Telnet client and enable the EnOcean listener. This displays information about the wireless data packets the DataStream receives. Normally an EnOcean device has a push button to send a radio packet. Pressing this will flash the Status LED and display (among other things) the device ID.



Type	This integer specifies which value is read from the device. See earlier table.
Units	This text value is used in creating the header value in the CSV file (15 characters)

*The following settings are **only** used for the Synetica Wi-CT electricity sensor*

Fuel Type	This is used to link the metered fuel to a fuel type look-up, so that CO_2 calculations can be made.
CT Volts	This integer is used to calculate the power value. When using the Wi-CT to monitor 3-Phase supplies, use the average voltage value per phase. For example enter 230 not 415 for UK voltages. **New from Version 1.5.0** This value can also use the **Modbus Reference Register** feature, as described in the Modbus section. Thus, you can use a value of MB1, so that the latest voltage reading from a Modbus meter is used to calculate the power.
CT Range	This integer (whole number) is used to specify the amperage range of the Current Transformer (CT) connected to the input. These values are: 5, 10, 20, 50, 100, 150, 500, 800, 1000, 1500 and 2000. It represents the current rating of the CT in Amps.

Wi-CT DIP Switch Settings

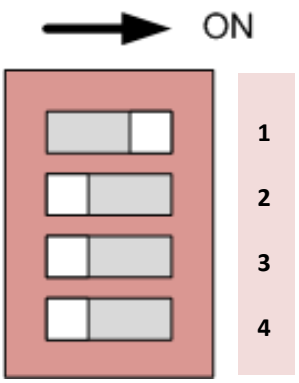
The DataStream can automatically select the 'CT Range' for each monitored Wi-CT. This is achieved by positioning the bit-switches inside the Wi-CT housing.

Switch 1 is the power switch and switches 2-4 tell the DataStream the current rating of the CTs that are attached. However, by setting the switches to the appropriate CT rating, as detailed below, the DataStream will automatically configure the scaling factors for the Wi-CT.

Example:

If the Wi-CT has 3x 50A CTs attached, set the DIP switches to: 2-OFF, 3-OFF, 4-ON

Note: *To enable the DataStream to use the DIP settings, the CT Range in the INI file must be set to 0 (zero) otherwise the value in the INI file will be used.*



Switch	Function
1:	Power
2-4:	CT Rating: see below

CT Rating	Switch		
	2	3	4
Manual	OFF	OFF	OFF
5A	**ON**	OFF	OFF
10A	OFF	**ON**	OFF
20A	**ON**	**ON**	OFF
50A	OFF	OFF	**ON**
100A	**ON**	OFF	**ON**
150A	OFF	**ON**	**ON**

Telnet Client Menu Options

Several options are available using the Telnet facility. These include changing the date and time, changing the root password and several diagnostic tools – depending on the model you have.

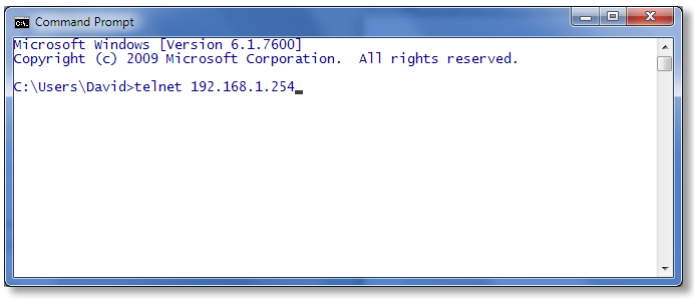
This example uses Microsoft Windows but any telnet client will work.

From a command window type

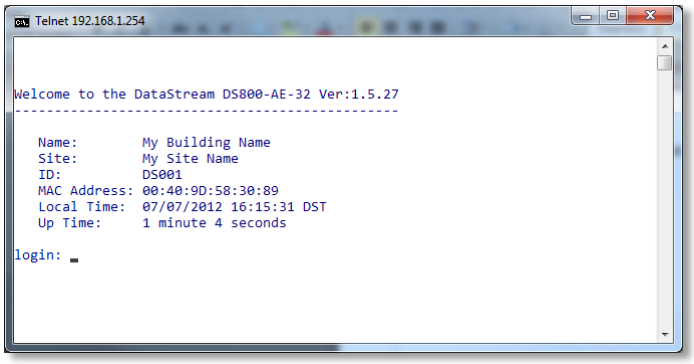
`telnet 192.168.1.254`

and press enter.

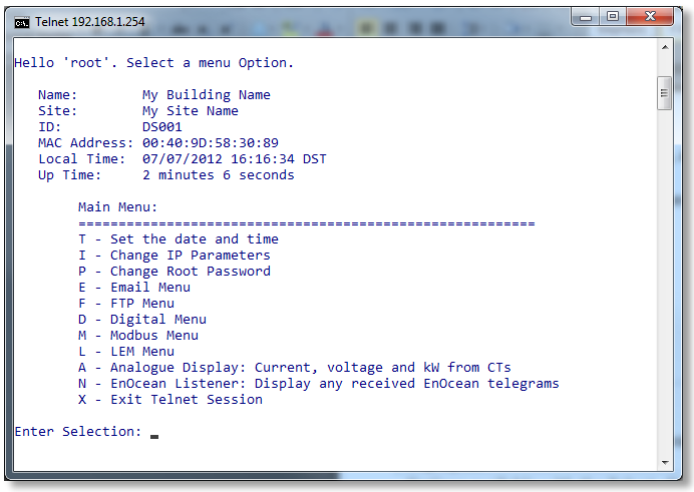
Substitute your DataStream's IP address if different.



Log in using **root** as the login name, and **password** as the password.
Use a different password if you have changed it.



And then select a menu from the available options displayed.



Options available with the Telnet client are:

```
T ........  Set the date and time
I ........  Change IP Parameters
P ........  Change Root Password
E ........  Email Menu
F ........  FTP Menu
D ........  Digital Menu
M ........  Modbus Menu
L ........  LEM Menu
A ........  Analogue Display: Current, voltage and kW from CTs
```

Only DataStream models with Analogue inputs enabled (e.g. DS800-A-32, DS800-AE-32)

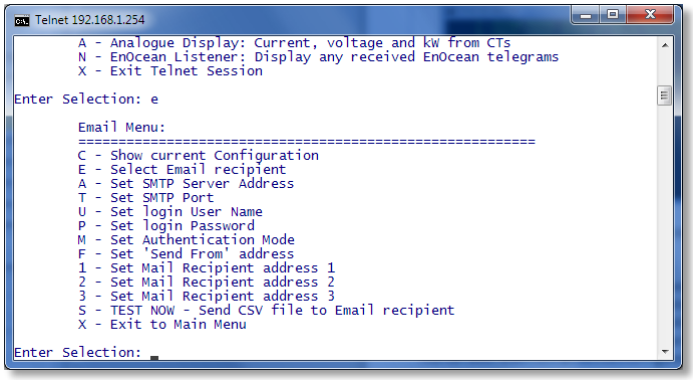
```
N ........  EnOcean Listener: Display any received EnOcean telegrams
```

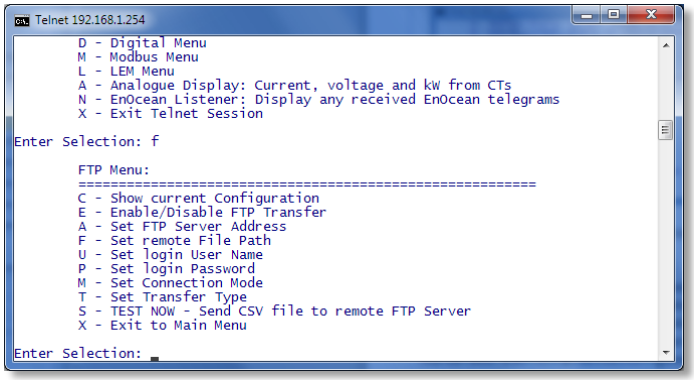
Only DataStream models with EnOcean wireless (e.g. DS800-E-32, DS800-AE-32)

```
X ........  Exit Telnet Session
```

Setting the date and time, IP address and changing the Root password are self-explanatory.

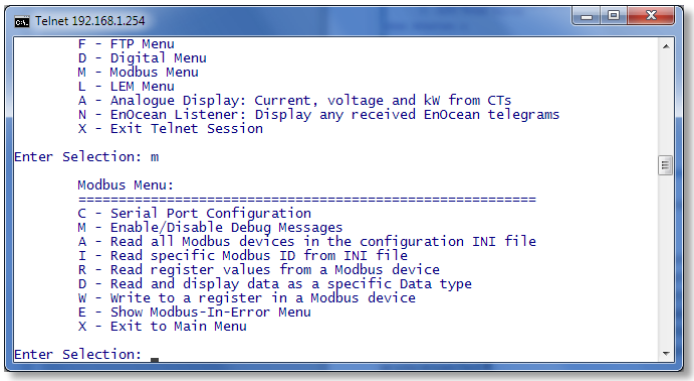
Email Menu and FTP Menu allows you to change all the INI settings from menus, and test your settings. Your changes are automatically saved in the INI file.

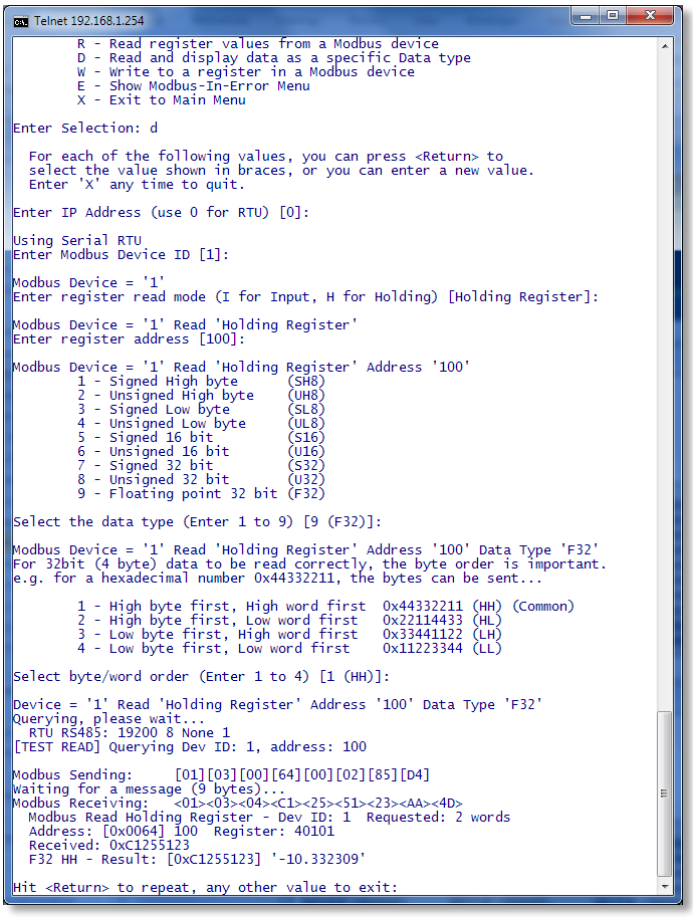




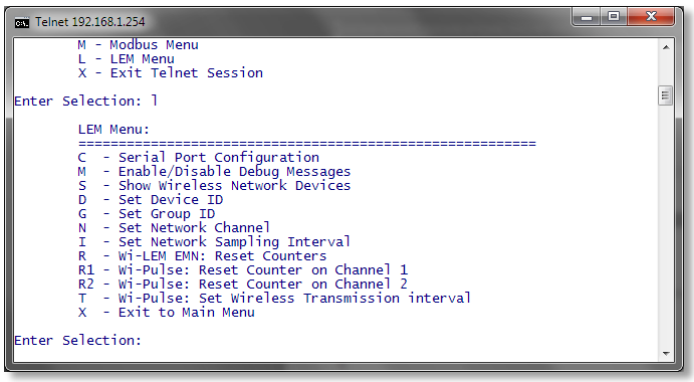
The Digital Menu lets you read the pulse count totals and virtual meter reading. You can also set the meter reading in the DataStream to match the actual meter.

The Modbus menu lets you configure the serial port settings and change debug message settings. You can also read and write from Serial RTU and Modbus TCP devices. All the menus are driven with options and from the user entering various values. You can also read from one or all devices configured in the INI file. This is useful to check that you have the INI file configured correctly.
Reading a value as a specific data type allows you to read, for example, a floating point value from a device.



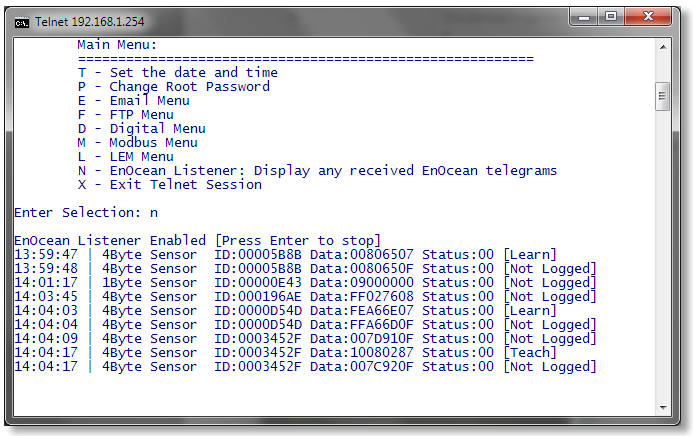


The LEM menu allows gives you diagnostic tools and menus to change many LEM specific options. This subject is very specific to LEM devices and is beyond the scope of this guide.



The Analogue Display will display 8 columns of readings, showing the amps from the connected CT, the voltage either from the INI file fixed value, or the last reading from a Modbus Meter and instantaneous kW reading from the attached current transformer. The kW reading is calculated based on the settings for each input – i.e. CT Volts and CT Range.

The EnOcean receiver built in to the DataStream is set to hi-sensitivity. When using the EnOcean Listener function, it has been seen that a device too close to the DataStream will not be detected as the radio signal is too strong. A distance of at least 2 metres is recommended.



Power up

On power up, the DataStream will initially illuminate the power LED. After a short delay, all eight Input LEDs will illuminate for a second then switch off. The DataStream then performs its boot-up procedure and the Input LEDs will flash and progress from 1 and 5, 2 and 6, 3 and 7, 4 and 8. Once the boot up procedure is complete, the lights will all switch off and they will function as normal – that is, light during a digital pulse input.
If, however all lights flash on and off, this indicates there was a problem reading the **config.ini** file. Details of the problem are logged in a file called **config_error.log** on the RAM storage. Use an FTP client to read the file to show the reason and on what line caused the fault. For example:

```
15/02/2012 12:23:56 Failed to parse INI file.
15/02/2012 12:23:57   Stopped at line: 63
15/02/2012 12:23:57     Expected Section: [Alert]
15/02/2012 12:23:58     Expected Item: 'SMTP Server'
15/02/2012 12:24:18 New default INI file created
15/02/2012 12:24:22 New INI file read completed
```

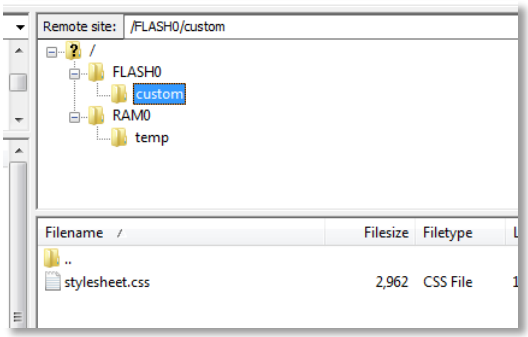
The DataStream renames the original file (that failed) and creates a working version. From this you should be able to find why the fault was caused.

The **Status** LED functions as an activity indicator (slow 2 second on/off pulses) and also flashes a very short pulse when a valid EnOcean data transmission is received (on an EnOcean enabled DataStream only).

Web Page Customisation

You can customise the Web Page style sheet. Create a folder on the FLASH0 storage area using an FTP Client called **custom**. Next, copy a style sheet into the folder. This must be called **stylesheet.css**.

This file is detected dynamically each time the web pages are requested from the DataStream. Thus you can make changes to the file and simply refresh your browser to see the results.



This is the text in the current Stylesheet.

```
BODY {
    FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif;
    FONT-SIZE: 12px;
    COLOR: #3F4926;
    background-image: url("Images/back.png");
    background-repeat: repeat-x;
}
SELECT {
    FONT-SIZE: 11px;
    FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif
}
INPUT {
    FONT-SIZE: 11px;
    FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif
}

TABLE, TR, TD, P, BR {
    FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif;
    FONT-SIZE: 12px;
}

.CONTENTHEADER {
    FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif;
    FONT-SIZE: 18px;
    FONT-WEIGHT: bold;
    COLOR: white;
    BACKGROUND-COLOR: #495929;
    PADDING: 8px;
    MARGIN: 0px;
}
```

```
.PAGETAB {
    FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif;
    FONT-SIZE: 12px;
    FONT-WEIGHT: normal;
    COLOR: #00008B;
    BACKGROUND-COLOR: #9CA859;
    BORDER-STYLE: solid;
    BORDER-COLOR: #9CA859;
    PADDING: 4px;
}
.PAGETAB-SELECTED {
    FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif;
    FONT-SIZE: 12px;
    FONT-WEIGHT: bold;
    COLOR: #008B00;
    PADDING: 4px;
}
PAGECONTENT {
    background-image: url("Images/content.png");
    background-repeat: repeat-x;
    BACKGROUND-COLOR: #F4FFD3;
}
.COPYRIGHT {
    FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif;
    FONT-SIZE: 10px;
}
.SMALLMESSAGE {
    FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif;
    FONT-SIZE: 10px;
    COLOR: #A0A0A0;
}
```

```
.DEADTAB {
    FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif;
    FONT-SIZE: 12px;
    FONT-WEIGHT: normal;
    COLOR: #00008B;
    BACKGROUND-COLOR: #3F4926;
    BORDER-STYLE: solid;
    BORDER-COLOR: #3F4926;
    PADDING: 4px;
}
A:HOVER {
    FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif;
    FONT-SIZE: 12px;
    COLOR: #00CC33;
    TEXT-DECORATION: underline;
}
.ERRORFIELD {
    FONT-WEIGHT: bold;
    BACKGROUND-COLOR: #FFDDDD;
    BORDER-COLOR: #E80000;
    BORDER-STYLE: solid;
}
.STATUSBOX {
    FONT-SIZE: 12px;
    FONT-WEIGHT: bold;
    COLOR: #404040;
    BACKGROUND-COLOR: #ffaa55;
    BORDER-COLOR: #ff8040;
    BORDER-STYLE: solid;
    BORDER-WIDTH: thin;
}
```

```
.BORDERCOLOR {
    BACKGROUND-COLOR: #495929;
}
.BORDER {
    PADDING: 0px;
    BORDER: 1px solid #495929;
}
.ERRORBOX {
    FONT-SIZE: 12px;
    FONT-WEIGHT: bold;
    COLOR: #E80000;
    BACKGROUND-COLOR: #FFDDDD;
    BORDER-COLOR: #E80000;
    BORDER-STYLE: solid;
    BORDER-WIDTH: thin;
}
.ACTIVE {
    COLOR: #FFFFFF;
    BACKGROUND-COLOR: #228B22;
}
.INACTIVE {
    COLOR: #FFFFFF;
    BACKGROUND-COLOR: #00008B;
}
```

Firmware Upgrades

Firmware upgrades are occasionally released by Synetica. These upgrades will contain feature enhancements and fixes that will improve the performance and functionality of the DataStream.

If this is required, Synetica will provide the necessary tools and instructions.

Frequently Asked Questions

How do I change the IP address?

There are several ways you can change the IP settings:

Using the web browser ..... On the Network Config page, the IP Address, subnet etc. are displayed in editable text boxes. Once you have made your changes and clicked [Apply] and the settings have been saved, you must reboot the DataStream for the changes to take effect. Use the [Reboot] web page.

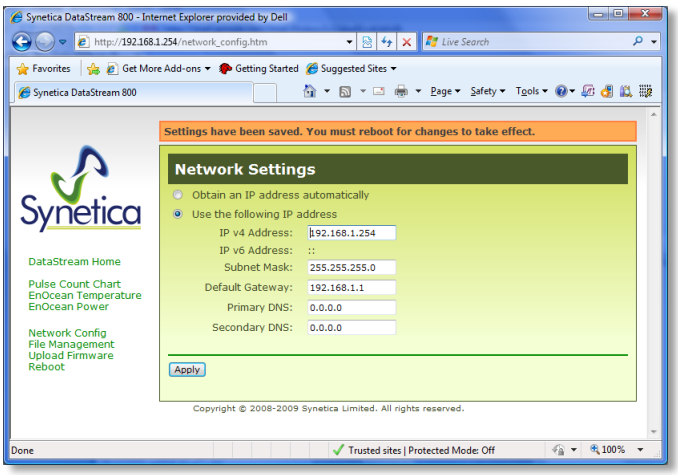
Using the telnet client ..... *New: for firmware version1.5.26*
From your telnet client, select the 'Change IP Parameters' menu. From here you can change enable or disable DHCP. If static addresses are used, you can change the IP address, subnet, gateway and DNS settings. Once set, the menu asks to confirm the settings and then the unit reboots for the changes to take effect.

Using the config.INI File ..... Edit the configuration file, and in the section [Network] you can change the IP Address, subnet etc. Once the configuration file has been saved on to the FLASH0 storage area, the DataStream will automatically reboot to apply the changes. *Do not switch the power off and on again using this method.* Allow the DataStream to restart itself – as it has to read the INI file and then copy these values to the non-volatile memory area used during re-starts.

Using the Console Port ..... Using a null-modem 9 pin D-Type cable, you can use a serial terminal application, for example HyperTerminal (on MS Windows) to connect to the DataStream (115200, 8, N, 1). When the DataStream reboots, it allows a small time period of 2 seconds for a key-press to be received on the serial console. Once a valid root password has been entered, the user may alter the IP address, subnet etc. The user may also change the root password from this menu.

The Network Config Web Page:

Remember that you must reboot the DataStream after you make any changes to the network settings.



How do I change the root password?

Using a Telnet client, log on to the DataStream. From the menu options, select the 'Change Root Password' option.

How do I change the name displayed in the web pages?

Open the configuration file. In the first section [Device], change the text in the Name item. This text is displayed in the top of the HTMP web pages, and the title of the Telnet client.

How do I find the device ID of an EnOcean sensor?

To discover an unknown device ID, use the Telnet client and enable the EnOcean listener. This displays information about the wireless data packets the DataStream receives. Normally an EnOcean device has a push button to send a radio packet. Pressing this will flash the EnOcean LED and display (among other things) the device ID. Remember you may have to keep the device 2 metres, or so, from the DataStream if it has a strong radio signal.

Appendix 1

Configuration file notes

These notes are laid out like the configuration file, but with various *comments* and notes to help with any sections not covered earlier in this document.

```
[Device]
Name = My Building Name
Site = My Site Name
; Name and Site are displayed on HTML web pages and used as a header in the Telnet sessions
ID = DS001
Description = Synetica DataStream DS800 Energy Data Logger
Log Interval = 30
; Log Interval: Valid values are 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60, 120, 180, 360, 720, 1440

[Network]
Use DHCP = 0
IP Address = 192.168.1.254
Subnet = 255.255.255.0
Gateway = 192.168.1.1
Primary DNS = 192.168.1.100
Secondary DNS = 0.0.0.0
HTTP Port = 80                              New for V1.0.25
Use HTTP Page Security = 1                  New for V1.0.25
Modbus TCP Slave Port = 502                 New for V1.0.25
Include Web Charts = 1                      New for V1.5.1

; HTTP Port: Changes the default TCP port of the Web Server
; Use HTTP Page Security: Use 1 or 0. When set to 0, will disable security on the
;  Home page and all chart pages. The 4 configuration pages at the bottom of the menu
;  list still require root access
; Modbus TCP Slave Port: When the DataStream is a Modbus TCP Slave, this is the IP port to connect to.
;  it can have data read from it by systems like a BMS or SCADA


[Time]
; Time Synchronisation uses UDP port 123
Use SNTP = 1
Primary IP = 152.2.21.1
Secondary IP = 198.82.1.201
; IP addresses only. Use a timeserver close to your own location.
TZ Sync Time = 18000
; Sync Time: frequency of contacting SNTP Server in seconds. Minimum is 30 seconds.
TZ STD Zone Offset = 0
TZ DST Zone Offset = 3600
; Offset times are in seconds
TZ DST Start Rule = 3.5.0/01:00:00
TZ DST End Rule = 10.5.0/02:00:00
; Month.Week.Day / Hour:Min:Sec
; See Appendix for further information - ; http://www.timeanddate.com/worldclock/
;
;===========================================

[Mail]
; SMTP Mail Send Configuration
SMTP Server = smtp.mail.com
; IP or FQDN (Fully Qualified Domain Name)
SMTP Port = 25
Authentication = 0
;    0 = MC_SMTP_AUTH_NONE
;    1 = MC_SMTP_AUTH_DIGESTMD5
;    2 = MC_SMTP_AUTH_CRAMMD5
;    3 = MC_SMTP_AUTH_LOGIN
;    4 = MC_SMTP_AUTH_PLAIN
;    5 = MC_SMTP_AUTH_ANY
;   [5] Checks the server ehlo response for the
;       server supported authentication in the
;       following order:
;         Digest-MD5
;         CRAM-MD5
;         LOGIN
;         PLAIN
Login Name =
Login Password =
Send From Address = datastream@synetica.net
```

```
[MailRecipient1]
Address = recipient1@synetica.net,recipient2@synetica.net,recipient3@synetica.net
; Ensure to separate multiple addresses with a comma and no spaces

[MailRecipient2]
Address = recipient4@synetica.net

[MailRecipient3]
Address = recipient5@synetica.net
; Each of the Mail Recipients, may be a comma separated list of email addresses, with up to
;  200 characters in total and up to 4 separate addresses. Ensure there are no spaces in the list.

;============================================

[FTP]
FTP Server = 192.168.1.10
; IP address or FQDN
Use Passive = 1
Use ASCII = 1
Login Name = ftpuser
Login Password = mypassword
File Path = /file

[Transfer]
Mail Recipient ID = 1
; Send logged data to mail address - use 0 to disable
Subject = Energy Usage File from $ID [$LT]
Body Text = Attached is the latest logged data from ID $ID on IP $IP. $CR$CRSynetica.$CR$CR
Use FTP = 1
Sync With Log Interval = 1
; If this is set to 1, it will transfer the CSV file after it has logged new values
; If it is set to 0, it will transfer the CSV file based on the next setting
Transfer Interval = 60
; This is used only when Sync With Log Interval is set to 0

;============================================

[Trace]
; Sends Email on start up with configuration file and any diagnostic files
Mail Recipient ID = 2
; Use 0 to disable
Severity Trigger = 1 [Not yet implemented]
Subject = DataStream System Trace Log
Body Text = Attached is the latest trace log.$CR$CRSynetica.$CR$CR

;============================================

[Config]
; Email configuration INI file (On Change)
Mail Recipient ID = 1
; Use 0 to disable
Subject = DataStream Configuration
Body Text = Attached is the latest configuration file.$CR$CRSynetica.$CR$CR

;============================================

[File]
Format = 1
; Set this to '0' (zero) to exclude the first data row.
; Use this when using the differential value logging with Modbus Meters. See Modbus Reference Register
Circular Log = 1
; If not circular log, logging stops when max lines complete
Max Line Count = 1440
Filename = EnergyData-$ID.csv
Include UTC = 1
Include Local = 1
UTC Header Text = UTCTimeStamp
Local Header Text = LocalTimeStamp
Time Format = %d/%m/%Y %T   ; See Time Format Codes

Include DI1 = 1
...
Include DI8 = 31
; DI: Digital Input. The number is 5 bit binary value that corresponds to:
; Bit 1: Consumption in Units per Log Interval [0000 0001] Decimal 1
;  Bit 2: Raw Pulse Count per Log Interval (and Total counts, that's 2 columns) [0000 0010] Decimal 2
;   Bit 3: Reading in Units (the Meter reading) - uses 'Decimal Numbers' as the precision [0000 0100] Decimal 4
;    Bit 4: kWh reading (required before CO2 calculation) [0000 1000] Decimal 8
;     Bit 5: Carbon Dioxide used in gram/kWh per Log Interval [0001 0000] Decimal 16
```

```
Include AI1 = 1
...
Include AI10 = 3
; AI: Analogue Input. The number is 2 bit binary value that corresponds to:
;.Bit 1: Consumption in kWh per Log Interval [0000 0001] Decimal 1
;..Bit 2: Carbon Dioxide used in gram/kWh per Log Interval [0000 0010] Decimal 2
; ** N O T E **
; If you select virtual meters 9 or 10, you must also include
; channels 1, 2, 3 (for 9) and/or 4, 5, 6 (for 10)

Include MB1 = 1
Include MB2 = 3
Include MB3 = 7
Include MB4 = 15
...
Include MB64 = 0
; MB: Modbus. The number is 5 bit binary value that corresponds to:
; Bit 1: Converted value in Units [0000 0001] Decimal 1
;  Bit 2: Raw reading (in raw data type) [0000 0010] Decimal 2
;   Bit 3: kWh reading (required before CO2 calc) [0000 0100] Decimal 4
;    Bit 4: Carbon Dioxide used in gram/kWh per Log Interval [0000 1000] Decimal 8
;     Bit 5: Differential Value (Interval) value [0001 0000] Decimal 16. See Modbus Reference Register

Include EnOcean1 = 1
...
Include EnOcean64 = 0
; EnOcean: The number is 4 bit binary value that corresponds to:
; Bit 1: Converted Value depending on device type [0000 0001] Decimal 1
;  Bit 2: Converted value in kWh - for Wi-CT power only [0000 0010] Decimal 2
;   Bit 3: Carbon Dioxide used in gram/kWh per Log Interval - for Wi-CT power only [0000 0100] Decimal 4
;    Bit 4: Radio Packet count per interval [0000 1000] Decimal 8

;===============================================

[Modbus RTU Master]
Port = 1
; Port: 0 - Off, 1 - RS485, 2-Bus RS232
Baud = 19200
; Baud: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
Parity = None
; Parity: None, Odd, Even
Data Bits = 8
; Data Bits: 5, 6, 7, 8
Stop Bits = 1
; Stop Bits: 1, 2
Retries = 3
; Number of retries when trying to read from devices – new from version 1.5.25
Timeout = 2
; Number of seconds to wait before timeout occurs when trying to read from devices – new from version 1.5.38
; Enter a value between 1 and 20.
Debug = 0
; Enables debug messages on Console port and Telnet sessions 0 = disabled, 1 = enabled, 2 = extra UART values

;===============================================

[DI1]      Digital Inputs
Name = Gas
Description = Main Gas Incomer
Units Multiplier = 0.1
Units = cu metres
Initial Reading = 8256.50
Max Reading = 99999.99
Rollover Count = 0
Last Rollover Date =
Decimal Numbers = 2
Fuel Type = 1

  Etc... to [DI8]

;===============================================
```

```
[AI1]       Analogue Inputs
Name = CT 1
Description = CT Meter 1
CT Volts = MB1
; CT Volts can use a fixed value, e.g. '230' or a Modbus Register Reference value for CT Volts
CT Range = 100
Fuel Type = 3

  [AI2] ... [AI10]
; AI9 and AI10 are 'virtual' 3 phase meters
; The virtual meter ignores the ct-range or ct-volts, it calculates the values from the source meters
;  Thus, [AI9] uses the values from [A1], [A2] and [A3] to calculate the 3-phase sum
;  and, [AI10] uses the settings from [A4], [A5] and [A6]


[MB1]       Modbus
IP = 0
; IP: Use 0 for RTU register, or a valid IP address for a TCP device
; New for V.1.5.21: Use -1 to use RTU Serial Port 1 or use -2 to use RTU Serial Port 2. Note: when
;  using these device-specific parameters, the baud rate and other serial parameters are taken from
;  the [Modbus RTU Master] section. These new options allow Modbus devices to be connected to both the
;  RS232 and RS485 port.

Port = 1502
; This is the IP port used to connect to the Modbus TCP device
ID = 1
; Modbus Device ID. Range 1 - 247
Template = 1
Name = Main Electric Meter
; Name used in header for CSV file
Description = Main Meter in Switch Room 1

  Etc... to [MB32] or [MB64] or [MB128] depending on model

;===============================================

[ModbusTemplate1]
;Name = Export kWh on MM101 meter
; ';Name' comment used for information only
Type = 3
; Type: 1, 2, 3, 4
; Coil Status, Input Status, Holding Register, Input Register
; Only types 3 and 4 supported
Register Address = 100
Data Type = U32
; S16 U16 S32 U32 SH8 UH8 SL8 US8 F32 Bit0 - Bit15
Word Order = HH
; Word/Byte order for 16 and 32 bit data types
Pre Offset = 0
Units Multiplier = 0.01
Post Offset = 0
Units = kWh
Fuel Type = 3
  Etc... up to max of [ModbusTemplate16] or [ModbusTemplate32] or [ModbusTemplate64]

;===============================================

[EnOcean1]
ID = 00005BB8
; ID: Device Hex ID range 00000000 - FFFFFFFF
Template = 1
Name = Room Temperature
; Name used in header for CSV file
Description = SR04 Sensor
  Etc... up to max of [EnOcean32] or [EnOcean64] or [EnOcean128]

;===============================================
```

```
[EnOceanTemplate1]
;Name = Temperature Sensor
; ';Name' comment used for information only
Type = 0
; Type: 0 ... 21
; See earlier EnOcean section for details on Type values
Units = Deg C
Fuel Type =
CT Volts =
CT Range =

[EnOceanTemplate2]
Type = 1
Name = Wi-CT Channel 1 RED
Units = kWh
Fuel Type = 3
CT Volts = MB1
; CT Volts can use a fixed value, e.g. '230' or a Modbus Register Reference value for CT Volts
CT Range = 100

  Etc... up to max of [EnOceanTemplate16] or [EnOceanTemplate32] or [EnOceanTemplate64]

;============================================

; CO2 Multiplier is in kilograms of CO₂ per kWh
; can create up to 20 'Fuel Type' entries

[FuelType1]
Name = Gas
kWh Multiplier = 11.0964
; This is used to convert from the measured units to kWh. So, this is an example of converting from
;  cubic meters to kWh
CO2 Multiplier = 0.194

[FuelType2]
Name = Oil
kWh Multiplier = 1
CO2 Multiplier = 0.265

[FuelType3]
Name = Electric
kWh Multiplier = 1
CO2 Multiplier = 0.544
```

Appendix 2

Storage Guidelines

The available space for file storage will determine how many rows of data you can store in the CSV file. The other factors are the number of columns you want to store, and the rate at which you are logging.
The storage available is approximately 2MB for each of the file areas (RAM0 and FLASH0)
The DataStream stores the logged data in a database file called **datafile.dat** on FLASH memory and the CSV files are stored in the RAM area.

The main CSV file size depends on the number of columns (one for each reading that is logged) and rows (one per logging interval). The file size also depends upon the data that is logged. Storing short numbers like '12.34' or '1234.5' will take less space than longer numbers like '12345678.123456' or '1.234567'.

Suggested Maximum Rows

Use the chart below to help you judge the number of rows you could store. Remember that these are only guidelines, and using values close to the maximum.

These recommendations are for when the **Include Web Charts** option is selected.

Columns of data to Store	Logging period: 15 mins	Logging period: 30 mins
16 Columns	30 days (2880 rows)	60 days (2880 rows)
32 Columns	15 days (1440 rows)	30 days (1440 rows)
64 Columns	7 days (672 rows)	15 days (720 rows)
128 Columns	4 Days (384 rows)	7 Days (336 rows)

These recommendations are for when the **Include Web Charts** option is **NOT** selected.

Columns of data to Store	Logging period: 15 mins	Logging period: 30 mins
16 Columns	60 days (5760 rows)	120 days (5760 rows)
32 Columns	30 days (2880 rows)	60 days (2880 rows)
64 Columns	15 days (1440 rows)	30 days (1440 rows)
128 Columns	14 Days (1344 rows)	21 Days (1008 rows)

For example, a system logging 32 Modbus readings every 30 minutes for 30 days (1440 rows) created a CSV file of 600KB.
The CSV files for the charts were, on average 40KB. So the storage required for the chart files is, 32 x 40KB = 1280KB.
Total storage required: 1280 + 600 = **1820KB** or **1.82MB**

The DataStream is designed to send the data to an aM&T software system (or similar) at a regular interval, normally every half hour. The storage the DataStream offers is enough to buffer conditions such as network failures.

For logging data at faster logging intervals, the time taken to generate the CSV files can become more of a factor than the storage space available.

When the various meters and environmental sensors have been read, the values are stored in the database file. This typically takes less than a second. Once this is completed, the CSV file is recreated from new on the RAM file system. This can take from a few seconds to several minutes if there are many columns and many rows to save.
Once this has been completed, all the data files for the web-based charts are created. Again this may take several minutes.

The time taken can be viewed from the diagnostic output from the using the COMMS console port.

For example, in some recent tests, logging 64 columns of data from serial Modbus meters took around 10 minutes to write the CSV files for 720 rows.
Therefore, for 64 columns you can log at an interval rate of 10 minutes, so that gives a storage capacity of five days.
1 row every 10 minutes is 6 rows an hour, 144 rows per day. 720/144 = 5 days

Other tests for logging 128 columns with charts, the CSV files (main and charts) took 2 to 3 seconds per row. So, writing 720 rows takes around 30 minutes.

Include Web Charts – V1.5.1

A new feature in firmware Version 1.5.1 allows you to switch off the generation of the CSV files for all charts. This is a new option in the INI file: For the latest model that can store up to 128 columns we recommend to always switch this option off.

```
Include Web Charts = 1
```

Set the value to 0 (zero) to disable the web chart CSV files. This also removes the menu option(s) in the web page menu.

This option enables larger CSV files to be created, so effectively allowing longer periods of data to buffer. Using the same example above; 32 columns at a 30 minute logging interval creates a 600KB file for a 30 day period.
Disabling the charts effectively saves 1.2MB, enabling the file to be expanded to hold 3 times the data, or 90 days of data which is a 1.8MB CSV file.

Using the same logic, a 64 column file would manage 45 days as a maximum without charts.

Turning the Include Web Charts option off also affects the speed at which you can log data.

Also, logging 128 columns of Modbus data at 9600 baud takes around 20 / 22 seconds to read. If there are any errors with the Modbus (even intermittent CRC errors, or timeouts) this can affect the overall time.
Test show that logging 128 columns at 19200 baud takes around 10 to 11 seconds.

With no charts, rows in the main CSV file are written at around 0.21 rows a second. So, 1000 rows in around 3.5 minutes. Data-file size will be around 500kB, CSV will be around 1MB.

So, if you are logging 128 columns of data, it would be impractical to log at a rate of faster than 5 minutes and log more than, say, 1440 rows (5 hours).

Appendix 3

Substitution codes

You can use keywords to replace certain parameters in several sections of the INI configuration file. For example, suppose you wanted the subject of an email to include the ID of the DataStream device: You can use "Email from $ID". The $ID will be replaced with the actual ID from the INI file.
If you did want to use '$ID' or any other parameter keyword as actual text, you should use a double $. i.e. $$ID will be interpreted as $ID.
Note: The keywords are case sensitive use **upper case**.

Static Substitutions

Use these in the [File] Filename, and in email Subject and Body parameters

Name	Keyword
Device name	$DEV
Site name	$SITE
Device ID	$ID
Log Interval	$LOG
Description	$DESC
IP Address	$IP

Format Codes

Only allowed in BODY of emails, not allowed in the Subject

Name	Keyword
Carriage Return	$CR
Tabulation character	$TAB

Dynamic Substitutions

Body and Subject parameters of emails

Name	Keyword	Description
UTC Date UK Format	$UDUK	Formatted as DD/MM/YYYY
UTC Date US Format	$UDUS	Formatted as MM/DD/YYYY
UTC Time	$UT	Formatted as hh:mm:ss
Local Date UK Format	$LDUK	Formatted as DD/MM/YYYY
Local Date US Format	$LDUS	Formatted as MM/DD/YYYY
Local Time (Military Format)	$LT	Formatted as hh:mm:ss
UTC File Friendly Date	$UFFD	Formatted as YYYY-MM-DD
UTC File Friendly Time	$UFFT	Formatted as hh-mm-ss
Local File Friendly Date	$LFFD	Formatted as YYYY-MM-DD
Local File Friendly Time	$LFFT	Formatted as hh-mm-ss
UTC Year	$UYEAR	Four digits, YYYY
UTC Month	$UMON	Two digits MM
UTC Month Name Short format	$UMNS	Abbreviated month name (English only)
UTC Month Name	$UMN	Full month name (English)
UTC Day	$UDAY	Two digits DD
UTC Day Name Short format	$UDNS	Abbreviated weekday name (English)
UTC Day Name	$UDN	Full week day name (English)
UTC Hour	$UHOUR	Two digits HH
UTC Minute	$UMIN	Two digits MM
UTC Second	$USEC	Two digits SS
Local Year	$LYEAR	As above
Local Month	$LMON	"
Local Month Name Short format	$LMNS	"
Local Month Name	$LMN	"
Local Day	$LDAY	"
Local Day Name Short format	$LDNS	"
Local Day Name	$LDN	"
Local Hour	$LHOUR	"
Local Minute	$LMIN	"
Local Second	$LSEC	"

Appendix 4

Time format codes

Code	Description
%A	is replaced by the locale's full weekday name.
%a	is replaced by the locale's abbreviated weekday name.
%B	is replaced by the locale's full month name.
%b %h	is replaced by the locale's abbreviated month name.
%C	is replaced by the century (a year divided by 100 and truncated to an integer) as a decimal number (00-99).
%c	is replaced by the locale's appropriate date and time representation.
%D	is replaced by the date in the format %m/%d/%y.
%d	is replaced by the day of the month as a decimal number (01-31).
%e	is replaced by the day of month as a decimal number (1-31); single digits are preceded by a blank.
%F	is replaced by the date in the format %Y-%m-%d.
%G	is replaced by the ISO 8601 year with century as a decimal number.
%g	is replaced by the ISO 8601 year without century as a decimal number (00-99).
%H	is replaced by the hour (24-hour clock) as a decimal number (00-23).
%I	is replaced by the hour (12-hour clock) as a decimal number (01-12).
%j	is replaced by the day of the year as a decimal number (001-366).
%k	is replaced by the hour (24-hour clock) as a decimal number (0-23); single digits are preceded by a blank.
%l	is replaced by the hour (12-hour clock) as a decimal number (1-12); single digits are preceded by a blank.
%M	is replaced by the minute as a decimal number (00-59).
%m	is replaced by the month as a decimal number (01-12).
%n	is replaced by a newline.
%p	is replaced by the locale's equivalent of either AM or PM.
%R	is replaced by the time in the format %H:%M.
%r	is replaced by the locale's representation of 12-hour clock time using AM/PM notation.
%S	is replaced by the second as a decimal number (00-60).
%s	is replaced by the number of seconds since the Epoch (00:00:00 UTC on January 1, 1970).
%T	is replaced by the time in the format %H:%M:%S.
%t	is replaced by a tab.
%U	is replaced by the week number of the year (Sunday as the first day of the week) as a decimal number (00-53).
%u	is replaced by the weekday (Monday as the first day of the week) as a decimal number (1-7).
%V	is replaced by the week number of the year (Monday as the first day of the week) as a decimal number (01-53). If the week containing January 1 has four or more days in the new year, then it is week 1; otherwise it is week 53 of the previous year, and the next week is week 1.
%W	is replaced by the week number of the year (Monday as the first day of the week) as a decimal number (00-53).
%w	is replaced by the weekday (Sunday as the first day of the week) as a decimal number (0-6).
%X	is replaced by the locale's appropriate time representation.
%x	is replaced by the locale's appropriate date representation.
%Y	is replaced by the year with century as a decimal number.
%y	is replaced by the year without century as a decimal number (00-99).
%%	is replaced by a single %.

Appendix 5

Time zone Information

The time zone (TZ) settings configure the DataStream to use the local time zone.

Both **TZ STD Zone Offset** and **TZ DST Zone Offset** values are entered as seconds relative to UTC (GMT).
For example, 8:00 hours difference = (8 x 60 x 60) = 28800
For example, 9:30 hours difference = (9 x 60 x 60) + (30 x 60) = 34200

These settings also configure the daylight saving time rule. The **TZ DST Start Rule** is used to indicate when the standard time zone changes to daylight saving time. The **TZ DST End Rule** is used to indicate when daylight saving time ends. Use this format:

"month.week.day/hour:minute:second"
where **day** >= 0 (Sunday) and <= 6 Saturday
week >= 1 (1st week) and <= 5 (5th week or last week) of the month
month >= 1 (January) and <= 12 (December)
hour >= 0 and <= 24; **minute** >= 0 and <= 59; **second** >= 0 and <= 59

For example, **4.1.0/02:00:00** indicates Sunday, the first week of April at 2:00 AM.

Note: The DST End Rule time is based on the current local time (DST time), not the STD time. If you look at www.worldtimezone.com you can see that the European times are given as UTC not DST which can confuse.
Also look at http://www.timeanddate.com/worldclock for reference.

TZ Sync Time is the frequency of contacting SNTP Server in seconds. The default is 18000 seconds, or 5 hours.

When looking for IP addresses of NTP Servers, look for servers that are geographically close to you and are Stratum 2 or 3 servers. For example, search the internet with a phrase like "*stratum 2 ntp time servers my_location"*
Other Sources: http://tf.nist.gov/tf-cgi/servers.cgi

Here are some examples using the configuration file:

UK Time Zone
Greenwich Mean Time – GMT

```
[Time]
Use SNTP = 1
Primary IP = 152.2.21.1
Secondary IP = 198.82.1.201
TZ Sync Time = 18000
TZ STD Zone Offset = 0
TZ DST Zone Offset = 3600
TZ DST Start Rule = 3.5.0/01:00:00
TZ DST End Rule = 10.5.0/02:00:00
```

New York, USA
Eastern Standard Time – EST
Standard time zone: UTC/GMT -5:00 hours

```
[Time]
Use SNTP = 1
Primary IP = 64.90.182.55
Secondary IP = 128.59.64.60
TZ Sync Time = 18000
TZ STD Zone Offset = -18000
TZ DST Zone Offset = -14400
TZ DST Start Rule = 3.2.0/02:00:00
TZ DST End Rule = 11.1.0/02:00:00
```

Adelaide, South Australia
Australian Central Standard Time – ACST
Standard time zone: UTC/GMT +9:30 hours

```
[Time]
Use SNTP = 1
Primary IP = 129.127.28.4
Secondary IP = 129.127.40.3
TZ Sync Time = 18000
TZ STD Zone Offset = 34200
TZ DST Zone Offset = 37800
TZ DST Start Rule = 10.1.0/02:00:00
TZ DST End Rule = 4.1.0/03:00:00
```

Vancouver, Canada
Pacific Standard Time – PST
Standard time zone: UTC/GMT -8:00 hours

```
[Time]
Use SNTP = 1
Primary IP = 192.31.216.30
Secondary IP = 192.6.38.127
TZ Sync Time = 18000
TZ STD Zone Offset = -28800
TZ DST Zone Offset = -25200
TZ DST Start Rule = 3.2.0/02:00:00
TZ DST End Rule = 11.1.0/02:00:00
```

Other time servers: 192.5.5.250