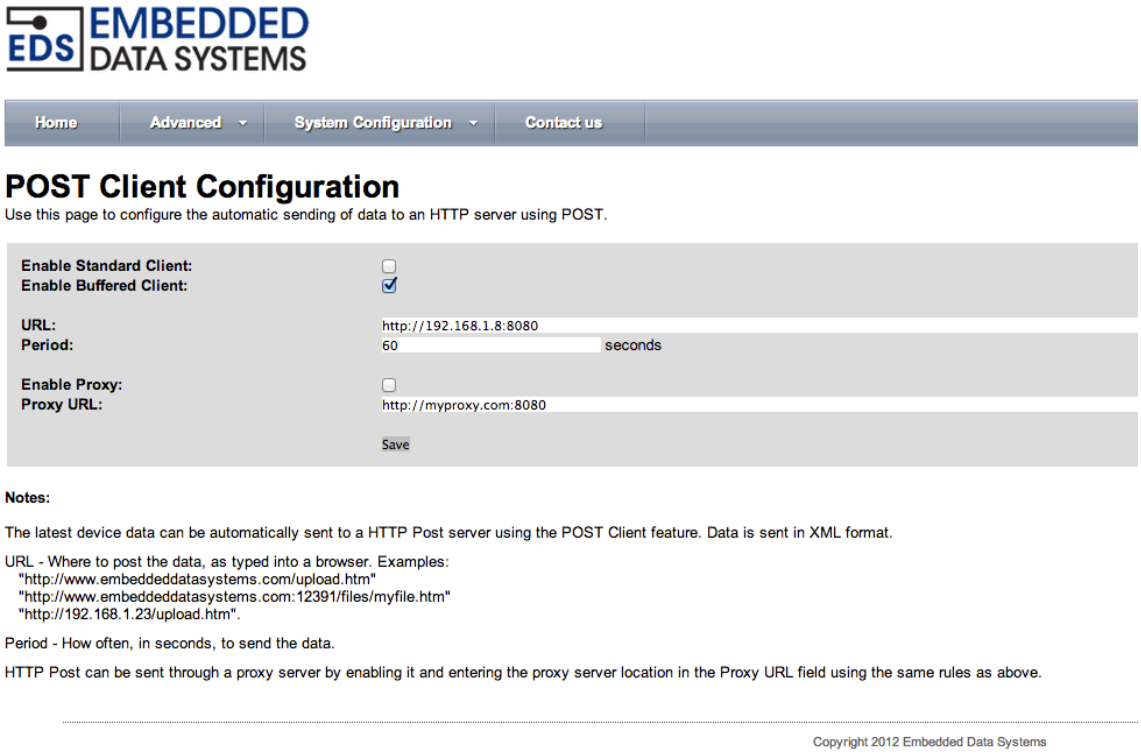
EDS Post Server – Java Version 1.00

Operation

This software creates a TCP server and listens for connections on the programmed port. It is designed to receive XML data pushed from one or a number of OW Servers and MeshNet Controllers. To operate, the OW Server and / or MeshNet controller must be programmed correctly, and the software must be running.

MeshNet Controller

To program the MeshNet controller, enter the DNS name or IP address in the Post Client page as follows:



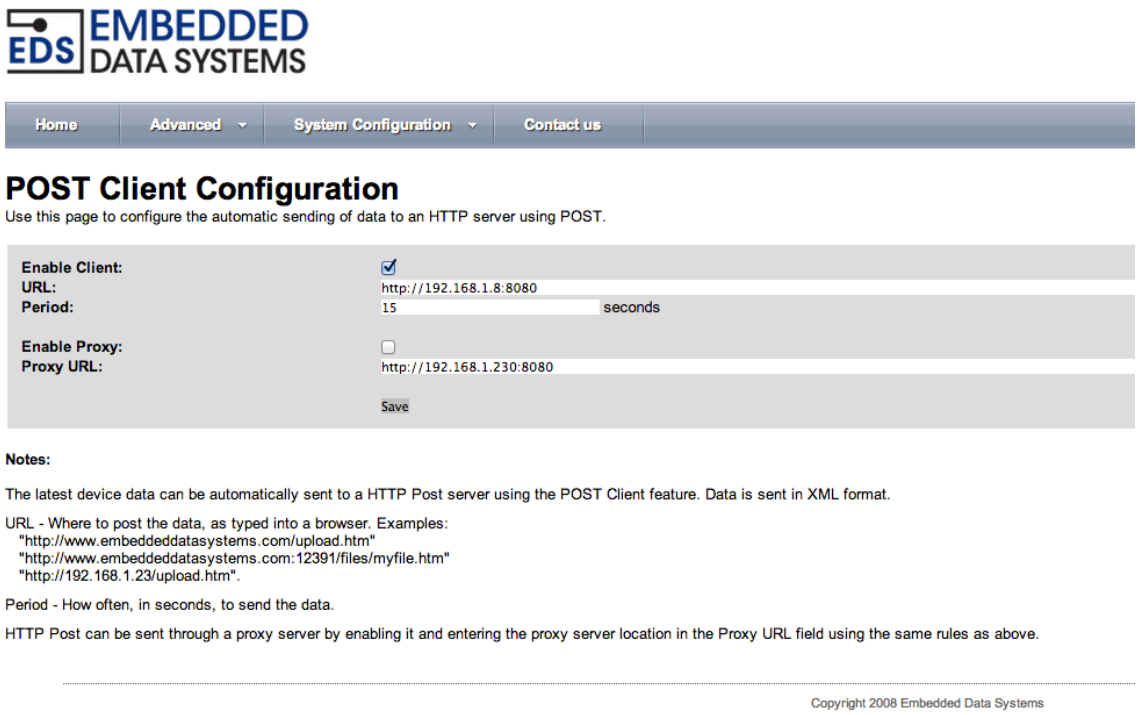
Note the following:

- One of the sending types is selected, in this case “Enable Buffered Client”. Both may be selected.
- The “URL” is set to the address of the server and the port is set to the same setting as in the program (see below). The “URL” field must begin with “http://”.
- The “Period” does not apply to “Buffered Client” sending, data is forwarded to the server within seconds of being received.
- After the changes are made, be certain to click the “Save” button.

OW Server v2

Enter the data as shown in the screen shot below.
Note the following:

- “Enable Client” has been selected.
- “URL” is programmed for the address and port of the server and begins with “http://”
- “Period” is set as desired.
- Do not forget to click “Save” when complete.

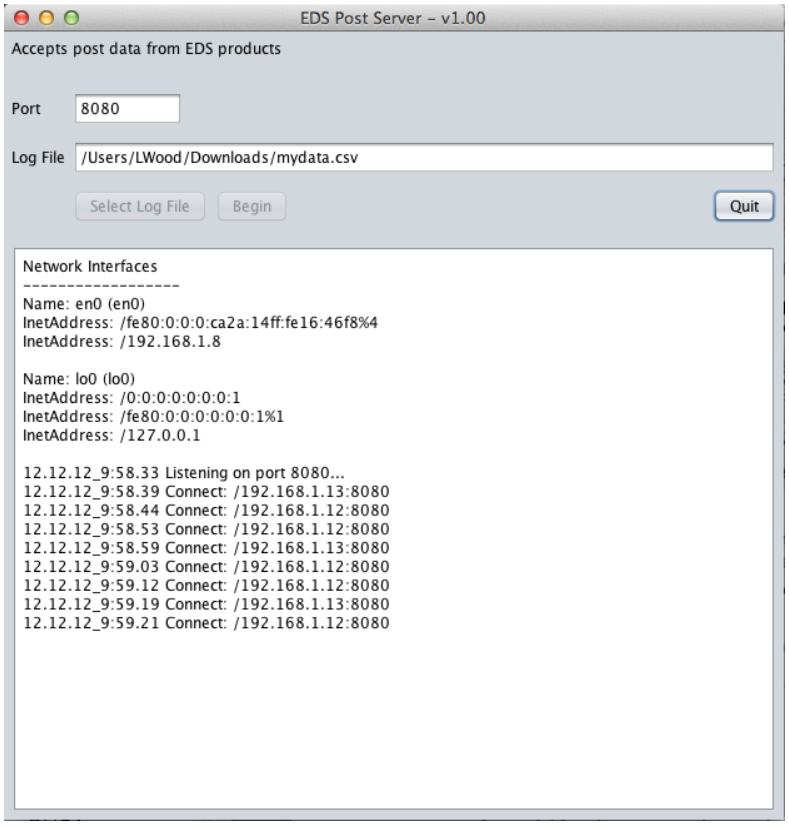


EDS Post Server

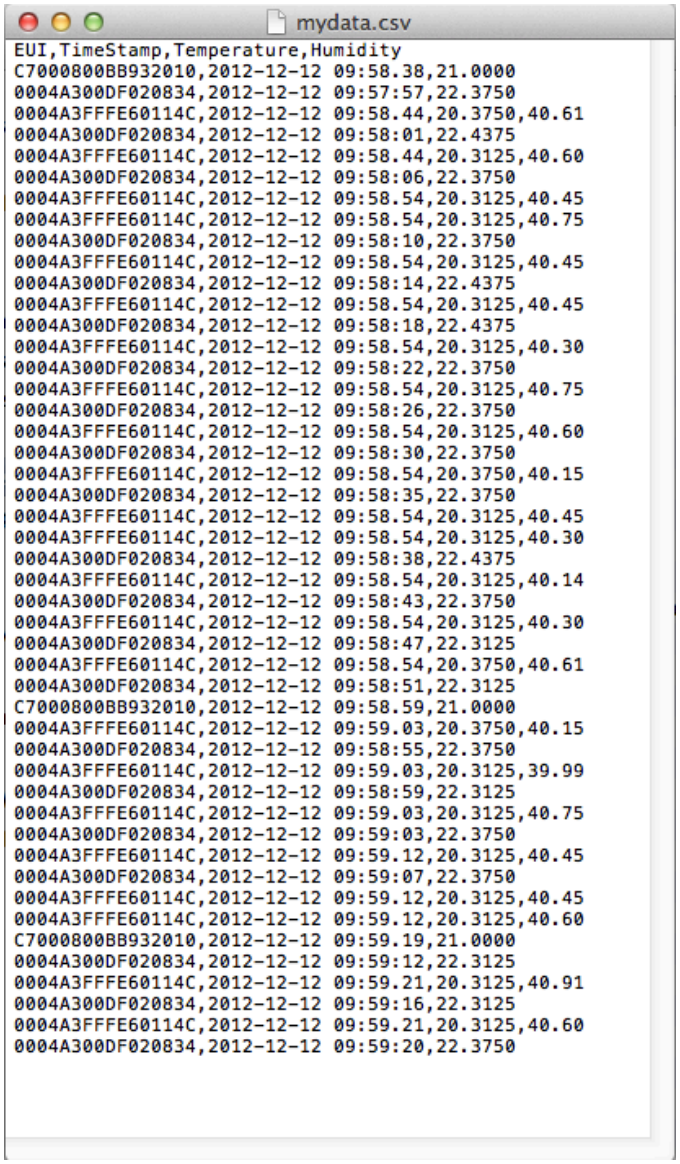
Start the post server program. This program is called "EDS_Post_Server.jar". Verify the port is set as desired (the above examples used the default of 8080). A log file may be selected, if a file is selected, one line per device record is written to the file. In the example below the log file is "mydata.csv" and is stored in "/Users/LWood/Downloads/". Begin listening for data by clicking the "Begin" button.

The information window at the bottom displays the local network ports available on the computer, then begins listening. In the example below the computer has 2 ports, 192.168.1.8 (this is the Ethernet port) and 127.0.0.1 (this is the loopback port). In this example, the OW Server v2 and MeshNet Controller are on the Ethernet network and so will send the data to 192.168.1.8 (see above).

When a connection is made it is displayed in the information window.



Below is an example of the log file. It consists of a list of the data as it has been received. Ideally this information would be sent to a database instead of a CSV file. Then is could be sorted and queried with ease.


mydata.csv

```
EUI,TimeStamp,Temperature,Humidity
C7000800BB932010,2012-12-12 09:58.38,21.0000
0004A300DF020834,2012-12-12 09:57:57,22.3750
0004A3FFFE60114C,2012-12-12 09:58.44,20.3750,40.61
0004A300DF020834,2012-12-12 09:58:01,22.4375
0004A3FFFE60114C,2012-12-12 09:58.44,20.3125,40.60
0004A300DF020834,2012-12-12 09:58:06,22.3750
0004A3FFFE60114C,2012-12-12 09:58.54,20.3125,40.45
0004A3FFFE60114C,2012-12-12 09:58.54,20.3125,40.75
0004A300DF020834,2012-12-12 09:58:10,22.3750
0004A3FFFE60114C,2012-12-12 09:58.54,20.3125,40.45
0004A300DF020834,2012-12-12 09:58:14,22.4375
0004A3FFFE60114C,2012-12-12 09:58.54,20.3125,40.45
0004A300DF020834,2012-12-12 09:58:18,22.4375
0004A3FFFE60114C,2012-12-12 09:58.54,20.3125,40.30
0004A300DF020834,2012-12-12 09:58:22,22.3750
0004A3FFFE60114C,2012-12-12 09:58.54,20.3125,40.75
0004A300DF020834,2012-12-12 09:58:26,22.3750
0004A3FFFE60114C,2012-12-12 09:58.54,20.3125,40.60
0004A300DF020834,2012-12-12 09:58:30,22.3750
0004A3FFFE60114C,2012-12-12 09:58.54,20.3750,40.15
0004A300DF020834,2012-12-12 09:58:35,22.3750
0004A3FFFE60114C,2012-12-12 09:58.54,20.3125,40.45
0004A3FFFE60114C,2012-12-12 09:58.54,20.3125,40.30
0004A300DF020834,2012-12-12 09:58:38,22.4375
0004A3FFFE60114C,2012-12-12 09:58.54,20.3125,40.14
0004A300DF020834,2012-12-12 09:58:43,22.3750
0004A3FFFE60114C,2012-12-12 09:58.54,20.3125,40.30
0004A300DF020834,2012-12-12 09:58:47,22.3125
0004A3FFFE60114C,2012-12-12 09:58.54,20.3750,40.61
0004A300DF020834,2012-12-12 09:58:51,22.3125
C7000800BB932010,2012-12-12 09:58.59,21.0000
0004A3FFFE60114C,2012-12-12 09:59.03,20.3750,40.15
0004A300DF020834,2012-12-12 09:58:55,22.3750
0004A3FFFE60114C,2012-12-12 09:59.03,20.3125,39.99
0004A300DF020834,2012-12-12 09:58:59,22.3125
0004A3FFFE60114C,2012-12-12 09:59.03,20.3125,40.75
0004A300DF020834,2012-12-12 09:59:03,22.3750
0004A3FFFE60114C,2012-12-12 09:59.12,20.3125,40.45
0004A300DF020834,2012-12-12 09:59:07,22.3750
0004A3FFFE60114C,2012-12-12 09:59.12,20.3125,40.45
0004A3FFFE60114C,2012-12-12 09:59.12,20.3125,40.60
C7000800BB932010,2012-12-12 09:59.19,21.0000
0004A300DF020834,2012-12-12 09:59:12,22.3125
0004A3FFFE60114C,2012-12-12 09:59.21,20.3125,40.91
0004A300DF020834,2012-12-12 09:59:16,22.3125
0004A3FFFE60114C,2012-12-12 09:59.21,20.3125,40.60
0004A300DF020834,2012-12-12 09:59:20,22.3750
```

Program Operation

The EDS_Post_Server program starts a thread to listen for socket connection requests. When a connection is requested, the 'Listener' thread starts a 'Receiver' thread, which receives the data from the OW Server or MeshNet Controller and separates the XML from the headers. The 'Receiver' then puts the XML data in a LinkedBlockingQueue. This queue is thread safe and operates as a FIFO. Many 'Receiver' threads can operate at the same time.

The EDS_Post_Server program also starts a 'Saver' thread, which takes the XML data from the LinkedBlockingQueue, extracts the desired information and saves it to a file if a "Log File" has been selected. It is in this routine that modifications can be made to save different data or save the data to a database instead of a file.

Software License

The EDS_Post_Server program and source is licensed as shown below.