Driver Manual

FS-8700-92 Lennox L Connection SBus

APPLICABILITY & EFFECTIVITY
Effective for all systems manufactured after May 2021.

Driver Revision: 1.03
Document Revision: 2.B

MSA Safety
1000 Cranberry Woods Drive
Cranberry Township, PA 16066 USA
Website: www.MSAsafety.com

U.S. Support Information:
+1 408 964-4443
+1 800 727-4377
Email: smc-support@msasafety.com

EMEA Support Information:
+31 33 808 0590
Email: smc-support.emea@msasafety.com

Contents

1 Description

The Lennox SBus Serial Driver allows the FieldServer to transfer data to and from devices over either RS-232 or RS-485 using Lennox SBus serial protocol. The FieldServer can emulate either a Server or Client.

FieldServer Mode	Nodes	Comments
Client	1	Only 1 Client node allowed on each port.
Server	31 (255)	Lennox product literature defines a limit of 31 nodes per network. (This limitation is assumed to be based on the limitations of the RS-485 interface). 255 nodes is the limit based on the address range permitted by the protocol.

1.1 Operating as a Client

This driver polls L Connection capable devices, reading status and other information. The retrieved data is loaded into two Data Arrays – one for status bits and one containing values. The location of each data element within the arrays is fixed. The locations are provided in the Data Array. The driver polls only one device per query – broadcasts (address zero) are not supported. The driver can poll the potential 255 devices per port. Set points, control data and operating modes can be written to the remote devices.

1.2 Operating as a Server

The driver can emulate up to 255 devices per port. Limited functionality is supported – the driver is only capable of responding with status and set point data. The Server responds with data which must be arranged in the FieldServer's Data Arrays in the format specified in the Data Array. No value validation is performed, therefore out of range values and illegal status/mode combinations are possible. The Server is responsible for validation before sending.

1.3 'Guest' Support

Guest devices may take over the Client functions. The driver will reclaim the master after some time period has expired. During the period where control is relinquished the Client driver will not poll any remote devices and data will not be updated.

The Client may be configured to operate as a guest or master. As a master it will poll devices as configured. As a guest it will attempt to become the master and if successful will poll devices for data for short periods and then relinquish control to the previous master and become idle (no new data) until it becomes the master again.

2 Driver Scope of Supply

2.1 Provided by the Supplier of 3rd Party Equipment

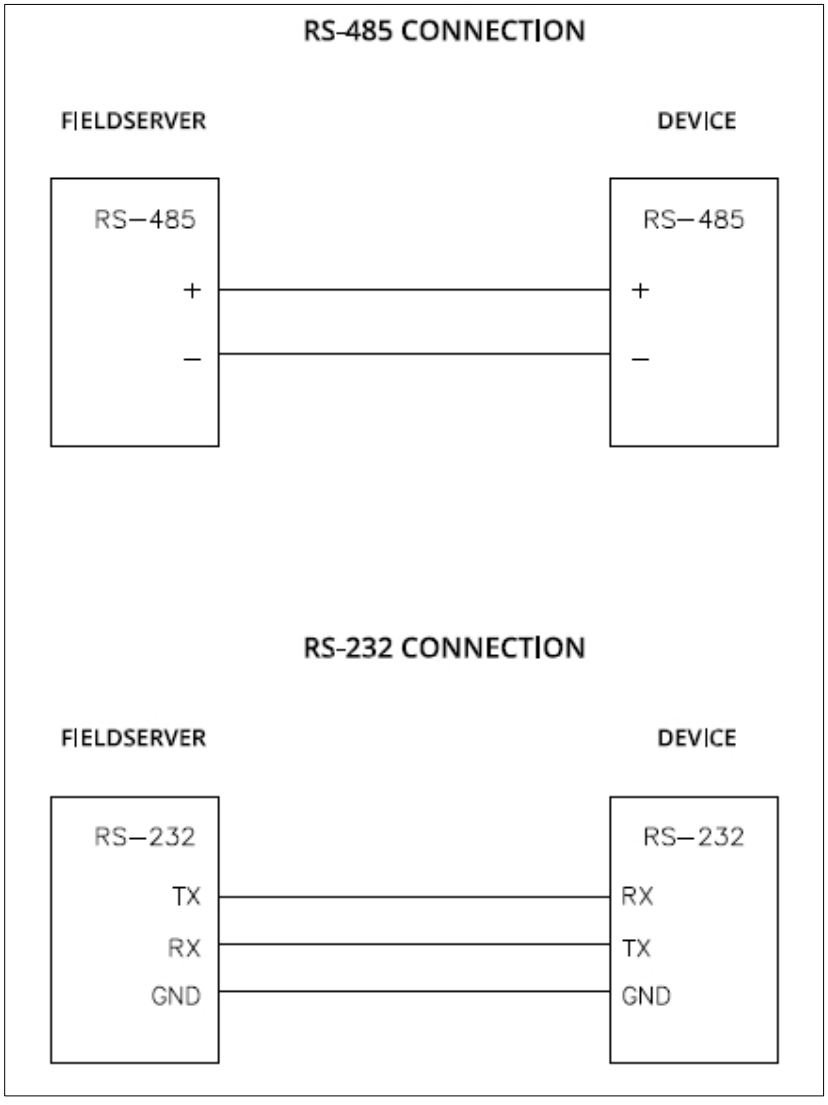
Part #	Description
	Lennox L-Series Controllers with SysBus Interface

3 Hardware Connections

The FieldServer is typically connected to the 'main' board mounted in/on the Lennox Unit's control panel. There are many different configurations of Lennox equipment. If unsure of the connection type, contact Lennox, quote the equipment model number and ask for a diagram providing connection information for the 'L Connection' or 'Sys Bus'.

Configure the Lennox device according to manufacturer's instructions. Ensure that the unit addresses are set correctly.

This typical connection diagram is based on fitted with either an A55 (M1) Main Control Board or a NTC1-1 (A113) control board interfacing to the FieldServer using RS-232.



NOTE:

- Typical L Connection (or Sys Bus) terminal block.
- Example based on a Lennox unit fitted with an A55 (M1) Main Control Board.
- Land shield on Ground Terminal of terminating device only.

Photo of a terminating connection (no additional devices connected) on a Lennox NTC1-1 Board. A number of Lennox control boards have S-Bus connections which are labeled in a similar way.

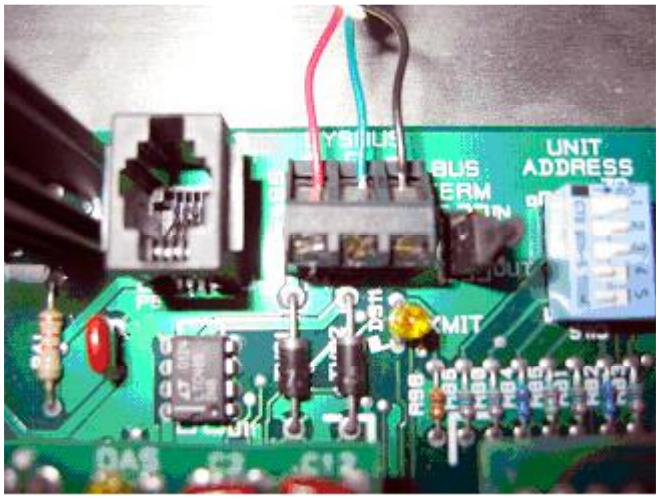


Photo of a terminating connection (no additional devices connected) to aA55 (M1) Main Control Board. The red wire is connected to the –ve terminal, the green wire to the 'G' terminal and the black wire to the +ve terminal. The Board manual says that the 'Bus Term' jumper must be in the out position for all units connected to the L Connection network.

3.1 Ground Wire / Cable Shield

The Lennox Corp recommend that the cable shield/drain should not be connected to the 'G' (ground) terminal on the L-Connection terminal block. On each network the cable shield of each cable segment should be joined to provide electrical continuity but must only be terminated at one end on the 'G' (ground) terminal of the board which terminates the network.

The installation instructions provided by the Lennox Corp for each control board should be followed. Some installation manuals identify cable type requirements. Consult with the equipment supplier if you have any questions.

3.2 Hardware Connection Tips / Hints

- Ensure that there are no duplicate nodes on the network.
- Ensure that no other unit on the network has been configured a bus master (active device polling other's for data).
- Ensure no device has been addressed as device zero.
- Some Lennox connection diagrams show that the GND wire on the Lennox control boards is not connected.
- Lennox product literature reports that the Sys Bus Transmit LED may blink as slowly as once every 30 seconds when many units are on the network.

4 Data Array Parameters

Data Arrays are “protocol neutral” data buffers for storage of data to be passed between protocols. It is necessary to declare the data format of each of the Data Arrays to facilitate correct storage of the relevant data.

Section Title		
Data_Arrays		
Column Title	**Function**	**Legal Values**
Data_Array_Name	Provide name for Data Array.	Up to 15 alphanumeric characters
Data_Array_Format	Provide data format. Each Data Array can only take on one format.	Float, Bit, Byte, Uint16, Uint32, Sint16, Sint32
Data_Array_Length	Number of Data Objects. Must be larger than the data storage area required by the Map Descriptors for the data being placed in this array.	1-10000

Example

```
//    Data Arrays
Data_Arrays
Data_Array_Name    , Data_Array_Format    , Data_Array_Length
DA_AI_01           , UInt16,              , 200
DA_AO_01           , UInt16               , 200
DA_DI_01           , Bit                  , 200
DA_DO_01           , Bit                  , 200
```

5 Client Side Configuration

For detailed information FieldServer configuration, please refer to the FieldServer Configuration manual. The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer (see ".csv" sample files provided with the FieldServer).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a Lennox SBus Server.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for Lennox SBus communications, the driver independent FieldServer buffers need to be declared in the "Data Arrays" section, the destination device addresses need to be declared in the "Client Side Nodes" section, and the data required from the servers needs to be mapped in the "Client Side Map Descriptors" section. Details on how to do this can be found below.

NOTE: In the tables below, * indicates an optional parameter and bold legal values are defaults.

5.1 Client Side Connection Descriptions

Section Title		
Connections		
Column Title	**Function**	**Legal Values**
Port	Specify which port the device is connected to the FieldServer.	P1-P2, R1-R2[1]
Protocol	Specify protocol used.	Lennox, Sbus, L-Connection
Baud*	Specify baud rate.	Vendor equipment supports 9600 baud. Driver supports 110 – 115200, standard baud rates.
Parity*	Specify parity.	**None** (vendor limitation)
Data_Bits*	Specify data bits.	**8**
Stop_Bits*	Specify stop bits.	**1**
Poll_Delay*	Time between internal polls.	0-32000 seconds, **1 second**

Example

```
//  Client Side Connections
Connections
Port    , Baud    , Parity   , Data_bits   , Stop_Bits   , Protocol   , Poll_Delay
R1      , 9600    , None     , 8           , 1           , Lennox     , 0.100s
```

[1] Not all ports shown may be supported by the hardware. Consult the instruction manual for the ports available.

5.2 Client Side Node Descriptors

Section Title		
Nodes		
Column Title	**Function**	**Legal Values**
Node_Name	Provide name for Node.	Up to 32 alphanumeric characters
Node_ID	Lennox device node address.	1-31(Driver supports 1-255)
Protocol	Specify protocol used.	Lennox, Sbus, L-Connection
Connection	Specify which port the device is connected to the FieldServer.	P1-P2, R1-R2[2]
Lennox_On_ Timeout_Behaviour*	On some Lennox devices the Node_ID can be changed on the fly using dip switches. If the Node_ID of a Lennox device operating as a Server is changed, the Client will receive no response to following polls sent to the old address. This generates a timeout. To override this default behavior, specify this parameter and set its value to poll_for_new_node_id. When there is a timeout, the driver sends a broadcast poll for the Node_ID. The driver will update its Node_ID based on the response from the Server and continue polling using the new Node_ID. If the server device does not respond to the broadcast poll or if the response is invalid or has errors, the driver reports a timeout to the kernel. If the timeout was not the result of the server's ID changing, the server will respond to the broadcast poll for the ID with the existing Node_ID. In this case the driver reports a timeout to the kernel.	**Timeout, poll_for_new_node_id**

Example

```
//    Consumer (Passive Client) Side Nodes
Nodes
Node_Name     , Node_ID     , Protocol     , Connection     , Node_Type
UNIT1         , 1           , Lennox       , R1             , Full
```

[2] Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

5.3 Client Side Map Descriptors

5.3.1 FieldServer Related Map Descriptor Parameters

Column Title	Function	Legal Values
Map_Descriptor_Name	Name of this Map Descriptor.	Up to 32 alphanumeric characters
Data_Array_Name	Name of Data Array where data is to be stored in the FieldServer.	One of the Data Array names from "Data Array" section above
Data_Array_Offset	Starting location in Data Array.	0 to maximum specified in "Data Array" section above
Function	Function of Client Map Descriptor.	Rdbc, Wrbc, Wrbx

5.3.2 Driver Related Map Descriptor Parameters

Column Title	Function	Legal Values
Node_Name	Name of Node to fetch data from.	One of the node names specified in the Node section
Data_Type	Data type	Register, Coil, AI, DI
Length	Length of Map Descriptor - dependent on the data being read. The driver validates the length specified and reports the length required if there is an error.	See **Section 7.1.1**
Address	This parameter has no meaning for the Lennox driver.	
Lennox_Func	Specify data to be read/written from the Lennox device. Specify either this field OR both Lennox_Cmd0/1 fields.	See **Section 7.1.1**
Lennox_Cmd0	These two parameters provide **alternate** methods of specifying data to be read/written from the Lennox device. The value may be specified in decimal or hexadecimal format. Hexadecimal format is more commonly used as all references to the parameter are quoted as hexadecimal values. Omit the Lennox_Func parameter if Lennox_Cmd0/1 parameters are specified.	0x01, 0x02 …. OR 1, 2 ,3 ….
Lennox_Cmd1		

5.3.3 Timing Parameters

Column Title	Function	Legal Values
Scan_Interval	Rate at which data is polled.	≥0.001s

5.4 Map Descriptor Examples

5.4.1 Simple Read

In this example a Map Descriptor is provided which reads all the available data from a Lennox Unit. The driver is required to send several different polls to the remote node. Each poll requests a different data set. The scan interval controls how often each poll is scheduled. Thus, even with a Scan_Interval of 1 second it will take the driver several seconds to completely read all the data.

Remember, each Map Descriptor is connected to a node by the Node_Name parameter and each node is in turn connected to a port; thus, each Map Descriptor is connected to a port. **Section 7.9** provides additional information.

```
//    Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name  , Data_Array_Name  , Data_Array_Offset  , Function  , Node_Name  , Length  , Scan_Interval  , Lennox_Func
MD_Read_All_01       , DA_Node1_Data     , 0                   , rdbc      , Node_1      , 950     , 1.0s             , Read Everything
```

In the above example:

- Map_Descriptor_Name – Giving Map Descriptors unique names is not essential but will help correct errors in the configuration if any exist.
- Data_Array_Name – The name of Data Array in which the data will be stored.
- Data_Array_Offset – Adjust where the driver stores data in the Data Array by changing this offset.
- Function – RDBC tells the driver to read the data continuously.
- Node_Name – The node name must match the name of a node defined in the Nodes Section. This is the name of the node which will be polled for data.
- Length – Data Array space reserved for storing data read from the remote node.
- Scan_Interval – Regularity at which the task is scheduled.
- Lennox_Func – This keyword tells the driver what data to read from the remote node.

5.4.2 Simple Write

In this example, the function is set to WRBX. This means that the write occurs on change/update. (Update means when the Data Array element is updated even if the value being stored is the same as it was before the store). When the 1st element of the Data Array called “DA_Node1_Control” is updated/changed then the write gets performed. The first element of the Data Array is watched because the Data_Array_offset is zero. The write occurs on change, so the Scan_Interval has no meaning, and the driver will ignore its setting. If the function is changed to WRBC, the write is performed on a periodic basis and the Scan_Interval is used to determine the period.

```
/   Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name   , Data_Array_Name    , Data_Array_Offset   , Function   , Node_Name   , Length   , Scan_Interval   , Lennox_Func
MD_Write_01           , DA_Node1_Control   , 0                   , Wrbx       , Node_1      , 1        , 1.0s            , Enable W1
```

In the above example:

- Data_Array_Name – When the driver sends a message to the remote Lennox device data from this array may be included in the message. **Section 7.2.1** indicates if the data is used. In this example the write ‘Enables W1’. This command does not require any data and hence the data from the Data Array is not included in the message. However, the data is used to trigger the write,
- Node_Name – The data gets sent to this node. The node name relates this Map Descriptor to a node definition where the address (Node_Id) is defined.
- Lennox_Func – What data gets sent to the remote Lennox device. The data set is determined by the keyword used to specify the ‘Lennox_Func’ parameter.

6 Server Side Configuration

For detailed information on FieldServer configuration, refer to the FieldServer Instruction Manual. The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer (see ".csv" sample files provided with the FieldServer).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a Lennox SBus Serial Client.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for Lennox SBus Serial communications, the driver independent FieldServer buffers need to be declared in the "Data Arrays" section, the FieldServer virtual node(s) needs to be declared in the "Server Side Nodes" section, and the data to be provided to the clients needs to be mapped in the "Server Side Map Descriptors" section. Details on how to do this can be found below.

NOTE: In the tables below, * indicates an optional parameter with the bold legal value as default.

6.1 Server Side Connection Parameters

Section Title
Connections

Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer.	P1-P2, R1-R2[3]
Baud*	Specify baud rate.	9600 (Vendor limitation)
Parity*	Specify parity.	None (Vendor limitation)
Data_Bits*	Specify data bits.	**8** (Vendor limitation)
Stop_Bits*	Specify stop bits.	**1** (Vendor limitation)
Protocol	Specify protocol used.	Lennox, Sbus, L-Connection[4]
Multidrop*	If you have configured the FieldServer as a Lennox Server and the FieldServer node is on a network with more than two nodes (one Server, one master) then enable Multidrop mode.	**Disable**/Enable

Example

```
//  Server Side Connections
Connections
Port    , Baud    , Parity    , Data_bits    , Stop_Bits    , Protocol
R1      , 9600    , None      , 8            , 1            , Lennox
```

[3] Not all ports shown may be supported by the hardware. Consult the instruction manual for details of the ports available.
[4] These three keywords are synonyms and can be interchanged without effect

6.2 Server Side Node Parameters

Section Title		
Nodes		
Column Title	**Function**	**Legal Values**
Node_Name	Provide name for node.	Up to 32 alphanumeric characters
Node_ID	Lennox device node address.	1-31 (Driver supports 1-255)
Protocol	Specify protocol used	Lennox, Sbus, L-Connection
Port*	If you have configured the FieldServer to emulate a single Lennox unit then you do not need to connect the node to a port as the FieldServer can respond to polls, irrespective of which port the poll arrived on. If, however, you configure the FieldServer to emulate two Lennox devices with the same Node_ID, each on a separate network then you must connect the node to a port.	P1-P2, R1-R2[5]

Example

```
//    Server Side Nodes
Nodes
Node_Name     , Node_ID     , Protocol     , Port
Unit1         , 1           , Lennox       , R1
```

6.3 Server Side Map Descriptor Parameters

6.3.1 FieldServer Specific Map Descriptor Parameters

Column Title	Function	Legal Values
Map_Descriptor_Name	Name of this Map Descriptor.	Up to 32 alphanumeric characters
Data_Array_Name	Name of Data Array where data is to be stored in the FieldServer.	One of the Data Array names from "Data Array" section above
Data_Array_Offset	Starting location in Data Array.	0 to maximum specified in "Data Array" section above
Function	Function of Client Map Descriptor.	Server
Server_Hold_Timeout*	Specifies time FieldServer will reserve Server side connection while waiting for the Client side to update data in Data Array (if necessary)	>1.0s

[5] Not all ports shown may be supported by the hardware. Consult the instruction manual for details of the ports available.

6.3.2 Driver Specific Map Descriptor Parameters

Column Title	Function	Legal Values
Node_Name	Name of Node to fetch data from.	One of the node names specified in “Server Node Descriptor” above
Data_Type	Data type	Register, Coil, AI, DI
Length	Length of Map Descriptor. The length is dependent on the data being read. The driver validates the length specified and reports the length required if there is an error.	See **Section** 7.1.1
Address	This parameter has no meaning for the Lennox driver.	
Lennox_Func	Specifies the data to be read/writen from the Lennox device. Specify this field OR both of the Lennox_Cmd0/1 fields.	See **Section** 7.1.1
Off_String	These parameters provide **alternate** methods of specifying data to be read/written from the Lennox device. The value may be specified in decimal or hexadecimal format. Hexadecimal format is more commonly used as all references to the parameter are quoted as hexadecimal values. Both parameters must also be specified and the Lennox_Func parameter omitted.	0x01, 0x02 …. OR 1, 2 ,3 ….
Length		

6.4 Map Descriptor Example

6.4.1 Provide Global Configuration Status Data

A file, primserv.csv is supplied with the Driver. This file has an example of every Server Map Descriptor required to define a fully capable Server. One Server Map Descriptor is required for each capability you wish to define for the Server.

In this example, one capability is illustrated – the ability of the Server to provide global configuration status data. This is done by setting the Lennox_Func parameter to the value "Send Expanded global variables status". If you don't define this Server Map Descriptor then when the Server receives a request for this information, the Server will respond with a negative acknowledgement indicating it could not provide the data. Server Map Descriptors are similar to Client Map Descriptors. The main difference lies in the 'Function' and the fact that Server MD's do not require a Scan_Interval as they define capabilities and not tasks.

```
/   Server Side Map Descriptors

Map Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Length , Lennox_Func
MD_Read_All_01      , DA_Node1_Data   , 0                 , Server   , Node_1    , 950    , Send Expanded global variables status
```

In the above example:

- Map_Descriptor_Name – Giving Map Descriptors unique names is not essential but will help correct errors in any configuration.
- Data_Array_Name – The name of the Data Array in which the data will be stored. Refer to **Section 7.2.1**.
- Data_Array_Offset – Adjust where the driver stores data in the Data Array by changing this offset.
- Function – Use 'Server' or 'passive' to define a Server side MD.
- Node_Name – The node name must match the name of a node defined in the Nodes Section. This is the name of the node which will be polled for data.
- Length – Data Array space reserved for storing data read from the remote node.
- Lennox_Func – This parameter defines the capability of the MD. The MD links the capability to a Data Array (by name, offset and length). This when the bus master polls for this status data, this Server can respond (because the MD exists) and driver knows where to fetch the data to send in the response. If this MD defines a capability to accept a command (write) from the bust master then the Map Descriptor tells the MD where to store the incoming data.

7 Advanced Topics

7.1 Commands and Capabilities (Lennox_Func Parameter)

The table below provides a list of keywords that can be used when specifying the Lennox_Func parameter.

For a Client: If you do not create a Map Descriptor with one of these keywords then the Client cannot execute that command. The scaling information determines how the driver converts the data read from the Data Array before packing it into the message. This is obviously only applicable for 'writes'.

For a Server: If you do not create a Map Descriptor with one of these keywords then the Server cannot respond to the equivalent poll from the Client. Scaling information is not relevant.

7.1.1 Lennox_Func Parameter Values

Keyword	Command Group ID	Command ID	Scaling Multiplier (y=mx+c)	Scaling Constant (y=mx+c)	Min MD Length	Max MD Length	Function	Device Category
--Special--Only for responses	0xff	0xff	1	0	1	1		1
Read Everything	0x77	0x77	1	0	1	950	Read	1
Sorted Alarm List	0x76	0x76	1	0	1	87	Read	1
Send current heating and cooling setpoints	0x01	0x00	1	0	1	4	Read	1
Current heating and cooling setpoints	0x01	0x00	1	0	1	1	Read	1
Set occupied heating setpoint to value	0x01	0x01	-4.0F	400.0F	1	1	Write	1
Set unoccupied heating setpoint to value	0x01	0x02	-4.0F	400.0F	1	1	Write	1
Set occupied cooling setpoint to value	0x01	0x03	-4.0F	400.0F	1	1	Write	1
Set unoccupied cooling setpoint to value	0x01	0x04	-4.0F	400.0F	1	1	Write	1
Send Reheat Status	0x01	0x05	1	0	1	1	Read	1
Reheat Status	0x01	0x06	1	0	1	1	Read	1
Set RH setpoint to value	0x01	0x07	2.5500F	0	1	1	Write	3
Current Setpoints	0x01	0x08	1	0	1	1	Read	1
Set heating and cooling remote DAT setpoints	0x01	0x0C	1	0	2	2	Write	2
Set single zone air temperature setpoint to value.	0x01	0x0E	-4	400.0F	1	1	Write	2
Set single zone air temperature setpoint offset to value	0x01	0x0F	4	0	1	1	Write	2
Set duct and building remote pressure setpoints to value	0x01	0x13	1	0	4	4	Write	2
Disable W1	0x01	0x20	1	0	1	1	Write	1
Disable W2	0x01	0x21	1	0	1	1	Write	1
Disable Y1	0x01	0x22	1	0	1	1	Write	1
Disable Y2	0x01	0x23	1	0	1	1	Write	1
Disable G	0x01	0x25	1	0	1	1	Write	1
Enable W1	0x01	0x28	1	0	1	1	Write	1
Enable W2	0x01	0x29	1	0	1	1	Write	1
Enable Y1	0x01	0x2A	1	0	1	1	Write	1
Enable Y2	0x01	0x2B	1	0	1	1	Write	1
Enable G	0x01	0x2D	1	0	1	1	Write	1
Change demands inputs to	0x01	0x2E	1	0	1	1	Write	1
Send Equipment Configuration	0x01	0x38	1	0	1	1	Read	1
Disable smoke input	0x01	0x40	1	0	1	1	Write	1
Disable occupied input	0x01	0x41	1	0	1	1	Write	3
Disable night setback override timer	0x01	0x42	1	0	1	1	Write	1
Disable economizer global input	0x01	0x43	1	0	1	1	Write	1
Disable warmup / cooldown mode	0x01	0x44	1	0	1	1	Write	1
Disable emergency heat (heat pump only)	0x01	0x45	1	0	1	1	Write	1
Disable room sensor remote control mode	0x01	0x46	1	0	1	1	Write	1
Disable blower on; goto blower auto	0x01	0x47	1	0	1	1	Write	1
Enable smoke input	0x01	0x48	1	0	1	1	Write	1
Enable occupied input	0x01	0x49	1	0	1	1	Write	3
Enable night setback override timer	0x01	0x4A	1	0	1	1	Write	1

Keyword	Command Group ID	Command ID	Scaling Multiplier (y=mx+c)	Scaling Constant (y=mx+c)	Min MD Length	Max MD Length	Function	Device Category
Enable economizer global input	0x01	0x4B	1	0	1	1	Write	1
Enable warmup / cooldown mode	0x01	0x4C	1	0	1	1	Write	1
Enable emergency heat (heat pump only)	0x01	0x4D	1	0	1	1	Write	1
Enable blower on; goto blower auto	0x01	0x4E	1	0	1	1	Write	1
Close economizer damper	0x01	0x50	1	0	1	1	Write	1
Open economizer damper	0x01	0x51	1	0	1	1	Write	1
Turn economizer exhaust fan off	0x01	0x52	1	0	1	1	Write	1
Turn economizer exhaust fan on	0x01	0x53	1	0	1	1	Write	1
Go to economizer auto mode	0x01	0x54	1	0	1	1	Write	1
Go to economizer override mode	0x01	0x55	1	0	1	1	Write	1
Set damper minimum position closed	0x01	0x56	1	0	1	1	Write	1
Set damper minimum position from minimum position control	0x01	0x57	1	0	1	1	Write	1
Enable direct control of the economizer damper	0x01	0x58	1	0	1	1	Write	1
Disable direct control of the economizer damper	0x01	0x59	1	0	1	1	Write	1
Set Economizer damper to value	0x01	0x5A	1	0	1	1	Write	1
Set minimum economizer damper to value	0x01	0x5B	1	0	1	1	Write	3
Set economizer outdoor air to value.	0x01	0x5C	1	0	1	1	Write	3
Change compressor / reversing valve outputs	0x01	0x60	1	0	1	1	Write	3
Change gas heat outputs	0x01	0x61	1	0	1	1	Write	1
Change electric heat outputs	0x01	0x62	1	0	1	1	Write	1
Configuration Command not found	0x01	0x80	1	0	1	1		3
Equipment configuration does not match IMC type.	0x01	0x81	1	0	1	1		1
Illegal Equipment configuration	0x01	0x82	1	0	1	1		3
Configuration Done	0x01	0x83	1	0	1	1		1
Equipment Data	0x01	0x84	1	0	1	1		1
Input command done	0x01	0x89	1	0	1	1		3
Illegal HVAC command	0x01	0x8A	1	0	1	1		3
EMI command done	0x01	0x8B	1	0	1	1		1
Setpoint changed	0x01	0x8C	1	0	1	1		3
Setpoint out of range	0x01	0x8D	1	0	1	1		1
Group HVAC Command	0x01	0x91	1	0	1	20	Write	1
Set AD input to override value	0x01	0x98	1	0	2	6	Write	2
Set emergency override to value	0x01	0x9A	1	0	1	1	Write	2
Set application mode to value	0x01	0x9C	1	0	1	1	Write	3
Set occupied manual override to value	0x01	0x9E	1	0	1	1	Write	2
Set occupied sensor override to value	0x01	0x9F	1	0	1	1	Write	2
Set maximum primary heating to value	0x01	0xAC	1	0	1	1	Write	2
Set maximum auxiliary heating to value	0x01	0xAD	1	0	1	1	Write	2
Set maximum compressor to use value	0x01	0xAE	1	0	1	1	Write	2
Set maximum dehumidification to value	0x01	0xAF	1	0	1	1	Write	2
Echo data	0x02	0x00	1	0	1	10	Write	3
Reply to echo	0x02	0x01	1	0	1	1	Write	3
Change address	0x02	0x5D	1	0	1	1	Write	1
Go online	0x02	0x5E	1	0	1	1	Write	3
Go Offline	0x02	0x5F	1	0	1	1	Write	3
Command Executed	0x02	0x62	1	0	1	1		3
Busy / retry for reply	0x02	0x63	1	0	1	1		3
Mode change executed	0x02	0x64	1	0	1	1		3
SBus message length error	0x02	0x65	1	0	1	1		3
Expanded device list	0x02	0x66	1	0	1	1		3
Device copyrights	0x02	0x67	1	0	44	88		2
Request for device ID	0x02	0x90	1	0	1	1	Read	3
Request for expanded device ID	0x02	0x91	1	0	1	10	Read	3
Request for copyright	0x02	0x92	1	0	44	88	Read	2
Request for system mode	0x02	0x94	1	0	1	1	Read	1
Illegal command	0x02	0xAC	1	0	1	1		3
No command found in block	0x02	0xAD	1	0	1	1		3

Keyword	Command Group ID	Command ID	Scaling Multiplier (y=mx+c)	Scaling Constant (y=mx+c)	Min MD Length	Max MD Length	Function	Device Category
Bad command block	0x02	0xAE	1	0	1	1		3
Bad check sum	0x02	0xAF	1	0	1	1		3
Device offline	0x02	0xD7	1	0	1	1		3
Timers cleared	0x05	0x00	1	0	1	1		1
Device reply	0x05	0x10	1	0	1	1		1
Command not found	0x05	0x70	1	0	1	1		1
Clear timers	0x05	0x80	1	0	1	10	Write	1
Device response	0x05	0x90	1	0	1	1		1
System reset	0x05	0xF0	1	0	1	1	Write	3
Expanded global status	0x06	0x05	1	0	1	1		1
Command not found	0x06	0x0F	1	0	1	1		3
Error buffer cleared	0x06	0x20	1	0	1	1		1
Error data block D0	0x06	0x22	1	0	1	1		1
Error buffer pointer	0x06	0x25	1	0	1	1		1
Version number	0x06	0x40	1	0	1	1		3
EEPROM busy	0x06	0x4D	1	0	4	8		3
Gateway global status	0x06	0x73	1	0	56	56		2
Send gateway global status	0x06	0x7B	1	0	56	56	Read	2
Send Expanded global variables status	0x06	0x86	1	0	72	72	Read	1
Send error block 0	0x06	0xA0	1	0	29	29	Read	1
Send error block 1	0x06	0xA1	1	0	29	29	Read	1
Send error block 2	0x06	0xA2	1	0	29	29	Read	1
Send error buffer pointer	0x06	0xA7	1	0	1	1	Read	1
Clear error buffer	0x06	0xA8	1	0	1	10	Write	1
Send version number	0x06	0xC0	1	0	4	8	Read	3
Go to local thermostat control mode	0x08	0x00	1	0	1	1	Write	1
Go to room sensor control mode with no backup	0x08	0x10	1	0	1	1	Write	1
Go to room sensor control mode with local thermostat backup	0x08	0x11	1	0	1	1	Write	1
Go to room sensor control mode with return air sensor backup	0x08	0x12	1	0	1	1	Write	1
Go to return air sensor control mode with no backup	0x08	0x20	1	0	1	1	Write	1
Go to return air sensor control mode with local thermostat backup	0x08	0x21	1	0	1	1	Write	1
Go to remote thermostat control mode with no backup	0x08	0x80	1	0	1	1	Write	1
Go to remote thermostat control mode with local thermostat backup	0x08	0x81	1	0	1	1	Write	1
Go to remote thermostat control mode with return air sensor backup	0x08	0x82	1	0	1	1	Write	1
Go to remote thermostat control mode with room sensor backup	0x08	0x83	1	0	1	1	Write	1
Go to air-handler control mode	0x08	0xB0	1	0	1	1	Write	1
Go to remote test mode	0x08	0xD0	1	0	1	1	Write	1
Go to remote standby mode	0x08	0xD1	1	0	1	1	Write	3
Go to manufacturing test mode	0x08	0xF0	1	0	1	1	Write	1

Explanation of Device Categories:

1=Non Gateway Devices

2=Gateway device only

3=Both categories of device

7.2 Where Data Gets Stored

This driver stores data at fixed Data Array locations as described in **Section 7.2.1**.

When a response is received from a Lennox device the data is stored in the Data Array specified in the Map Descriptor. Usually this data consists of byte values, whose individual bits report different settings and status. If a second Data Array is defined for each Map Descriptor, the byte values will be stored in the primary Data Array, and the driver will unpack the byte values and store the bit values in the secondary Data Array. When you define the primary and secondary Data Arrays in the CSV file they can have any data format.

Use **Section 7.2.1** as follows:

- Browse the data descriptions to identify the data of interest.
- Look up the Offset into the Primary array/Secondary Data Array.
- Consider the scaling information in determining what values to expect – when a Lennox device returns data it sends a raw value. Before the Client side of the driver stores the values in a Data Array it automatically scales the data using y=mx+c where m and c are obtained from **Section 7.2.1**.

Notes to Section 7.2.1:

- Note 1 – This data is obtained by the following read "Send Expanded global variables status".
- Note 2 – This data is obtained by read requests other than "Send Expanded global variables status".
- Note 3 – This data is not provided by "Send Expanded global variables status" when the device type is 0x60.

7.2.1 Data Locations

2ndry Array Offset[6]	Primary Array Offset	Scaling Multiplier	Scaling Constant	Data Category	Data filled in by other responses	Data filled in by 0x0605 with 0x30<=D29<=0x5f	Data filled in by 0x0605 with D29=0x60
0	0	1	0	Input Data Byte 3	Note 1	24VAC #1 Present	Note 3
1	0	1	0		Note 1	Option 1 input present	Note 3
2	0	1	0		Note 1	24VAC #2 Present	Note 3
3	0	1	0		Note 1	Not Used	Note 3
4	0	1	0		Note 1	Not Used	Note 3
5	0	1	0		Note 1	Air Flow Present	Note 3
6	0	1	0		Note 1	Not Used	Note 3
7	0	1	0		Note 1	Not Used	Note 3
8	1	1	0	Input Data Byte 2	Note 1	Low pressure 1 tripped	Note 3
9	1	1	0		Note 1	High pressure 1 tripped	Note 3
10	1	1	0		Note 1	Freeze stat 1 tripped	Note 3
11	1	1	0		Note 1	Defrost temperature switch 1 closed	Note 3
12	1	1	0		Note 1	Defrost pressure switch 1 closed	Note 3
13	1	1	0		Note 1	Dirty Filter	Note 3
14	1	1	0		Note 1	Not Used	Note 3
15	1	1	0		Note 1	Not Used	Note 3
16	2	1	0	Input Data Byte 1	Note 1	Primary Limit 1 Tripped	Digital input 0 high
17	2	1	0		Note 1	Secondary Limit 1 Tripped	Digital input 1 high
18	2	1	0		Note 1	Rollout switch 1 tripped	Digital input 2 high

[6] This data is only filled in by the driver if the Map Descriptor has the DA_Bit_Name parameter specified. An example is provided in **Section 7.4**.

2ndry Array Offset[6]	Primary Array Offset	Scaling Multiplier	Scaling Constant	Data Category	Data filled in by other responses	Data filled in by 0x0605 with 0x30<=D29<=0x5f	Data filled in by 0x0605 with D29=0x60
19	2	1	0		Note 1	Combustion blower 1 on	Digital input 3 high
20	2	1	0		Note 1	Gas valve sense 1 energized	Not Used
21	2	1	0		Note 1	Space occupied	Not Used
22	2	1	0		Note 1	Not Used	Not Used
23	2	1	0		Note 1	Not Used	Not Used
24	3	1	0	Input Data Byte 0	Note 1	G input	Dip switch 0 on
25	3	1	0		Note 1	W1 Input	Dip switch 1 on
26	3	1	0		Note 1	W2 input	Dip switch 2 on
27	3	1	0		Note 1	Y1 input	Dip switch 3 on
28	3	1	0		Note 1	Y2 input	Dip switch 4 on
29	3	1	0		Note 1	Smoke Detection Signal	Dip switch 5 on
30	3	1	0		Note 1	Not Used	Dip switch 6 on
31	3	1	0		Note 1	Not Used	Dip switch 7 on
32	4	1	0	Demand Inputs	Note 1	G demand	G demand
33	4	1	0		Note 1	W1 demand	W1 demand
34	4	1	0		Note 1	W2 demand	W2 demand
35	4	1	0		Note 1	Y1 demand	Y1 demand
36	4	1	0		Note 1	Y2 demand	Y2 demand
37	4	1	0		Note 1	Smoke detected	Not Used
38	4	1	0		Note 1	Space occupied	Space occupied
39	4	1	0		Note 1	Not Used	Not Used
40	5	1	0	Compressor Status	Note 1	Compressor 1 lockout	Note 3
41	5	1	0		Note 1	Compressor 2 lockout	Note 3
42	5	1	0		Note 1	Compressor 3 lockout	Note 3
43	5	1	0		Note 1	Compressor 4 lockout	Note 3
44	5	1	0		Note 1	Compressor 1 ignore strike 3	Note 3
45	5	1	0		Note 1	Compressor 2 ignore strike 3	Note 3
46	5	1	0		Note 1	Compressor 3 ignore strike 3	Note 3
47	5	1	0		Note 1	Compressor 4 ignore strike 3	Note 3
48	6	1	0	Compressor / Defrost Timers	Note 1	Compressor 1 on/off timer set	Note 3
49	6	1	0		Note 1	Compressor 2 on/off timer set	Note 3
50	6	1	0		Note 1	Compressor 3 on/off timer set	Note 3
51	6	1	0		Note 1	Compressor 4 on/off timer set	Note 3
52	6	1	0		Note 1	Defrost timer #1 on	Note 3
53	6	1	0		Note 1	Defrost timer #2 on	Note 3
54	6	1	0		Note 1	Not Used	Note 3
55	6	1	0		Note 1	Not Used	Note 3
56	7	-0.6792	164.45	Return Air Temp	Note 1	Return air temp	Return air temp
57	8	-0.6792	164.45	Discharge Air Temp	Note 1	Discharge Air Temp	Discharge Air Temp
58	9	7.843	0	Indoor Air Quality	Note 1	Indoor Air Quality	Indoor Air Quality
59	10	-0.636	131.56	Outdoor air temp	Note 1	Outdoor air temp	Outdoor air temp
60	11	-0.71	171.78	Room Air Temp (Wide Range)	Note 1	Room Air Temp (Wide Range)	Room Air Temp (Wide Range)
61	12	-0.25	100	Heating Setpoint	Note 1	Heating Setpoint	Heating Setpoint
62	13	-0.25	100	Cooling Setpoint	Note 1	Cooling Setpoint	Cooling Setpoint
63	14	1	0	Room Air Temp (Narrow Range)	Note 1	Room Air Temp (Narrow Range)	Not Used
64	15	1	0	Reset Status	Note 1	Reset Status	Not Used
65	16	1	0	Fan Status	Note 1	Fan #1 on	Not Used
66	16	1	0		Note 1	Fan #2 on	Not Used
67	16	1	0		Note 1	Fan #3 on	Not Used
68	16	1	0		Note 1	Fan #4 on	Not Used
69	16	1	0		Note 1	Fan #5 on	Not Used
70	16	1	0		Note 1	Fan #6 on	Not Used

2ndry Array Offset[6]	Primary Array Offset	Scaling Multiplier	Scaling Constant	Data Category	Data filled in by other responses	Data filled in by 0x0605 with 0x30<=D29<=0x5f	Data filled in by 0x0605 with D29=0x60
71	16	1	0		Note 1	Blower On	Blower On
72	16	1	0		Note 1	Service Relay on	Not Used
73	17	1	0	Compressor Status	Note 1	Compressor #1 on	Compressor #1 on
74	17	1	0		Note 1	Compressor #2 on	Compressor #2 on
75	17	1	0		Note 1	Compressor #3 on	Compressor #3 on
76	17	1	0		Note 1	Compressor #4 on	Not Used
77	17	1	0		Note 1	Reversing valve #1 on	Reversing valve #1 on
78	17	1	0		Note 1	Reversing valve #2 on	Reversing valve #2 on
79	17	1	0		Note 1	Not Used	Not Used
80	17	1	0		Note 1	Not Used	Not Used
81	18	1	0	Heat Status/Economizer Control	Note 1	Heating Stage #1 on	Heating Stage #1 on
82	18	1	0		Note 1	Heating Stage #2 on	Heating Stage #2 on
83	18	1	0		Note 1	Heating Stage #3 on	Not Used
84	18	1	0		Note 1	Heating Stage #4 on	Not Used
85	18	1	0		Note 1	Economizer damper fully closed	Not Used
86	18	1	0		Note 1	Turn on economizer Exhaust Fan	Not Used
87	18	1	0		Note 1	Open economizer damper to 100%	Not Used
88	18	1	0		Note 1	Economizer override	Not Used
89	19	1	0	Heating Stage #2 Input Status	Note 1	Primary limit tripped	Not Used
90	19	1	0		Note 1	Secondary limit tripped	Not Used
91	19	1	0		Note 1	Rollout switch limit tripped	Not Used
92	19	1	0		Note 1	Combustion Air Blower On	Not Used
93	19	1	0		Note 1	Gas valve On	Not Used
94	19	1	0		Note 1	Not Used	Not Used
95	19	1	0		Note 1	Not Used	Not Used
96	19	1	0		Note 1	Not Used	Not Used
97	20	1	0	Heating Stage #2 Input Status	Note 1	Primary limit tripped	Not Used
98	20	1	0		Note 1	Secondary limit tripped	Not Used
99	20	1	0		Note 1	Rollout switch limit tripped	Not Used
100	20	1	0		Note 1	Combustion Air Blower On	Not Used
101	20	1	0		Note 1	Gas valve On	Not Used
102	20	1	0		Note 1	Not Used	Not Used
103	20	1	0		Note 1	Not Used	Not Used
104	20	1	0		Note 1	Not Used	Not Used
105	21	1	0	Freezestat/Defrost Status	Note 1	Freezestat #1 tripped	Not Used
106	21	1	0		Note 1	Freezestat #2 tripped	Not Used
107	21	1	0		Note 1	Freezestat #3 tripped	Not Used
108	21	1	0		Note 1	Freezestat #4 tripped	Not Used
109	21	1	0		Note 1	Defrost Temp #1 switch closed	Not Used
110	21	1	0		Note 1	Defrost Temp #2 switch closed	Not Used
111	21	1	0		Note 1	Defrost pressure #1 switch closed	Not Used
112	21	1	0		Note 1	Defrost pressure #2 switch closed	Not Used
113	22	1	0	Pressure Status	Note 1	Compressor #1 low pressure ok	Not Used
114	22	1	0		Note 1	Compressor #2 low pressure ok	Not Used

2ndry Array Offset[6]	Primary Array Offset	Scaling Multiplier	Scaling Constant	Data Category	Data filled in by other responses	Data filled in by 0x0605 with 0x30<=D29<=0x5f	Data filled in by 0x0605 with D29=0x60
115	22	1	0		Note 1	Compressor #3 low pressure ok	Not Used
116	22	1	0		Note 1	Compressor #4 low pressure ok	Not Used
117	22	1	0		Note 1	Compressor #1 high pressure ok	Not Used
118	22	1	0		Note 1	Compressor #2 high pressure ok	Not Used
119	22	1	0		Note 1	Compressor #3 high pressure ok	Not Used
120	22	1	0		Note 1	Compressor #4 high pressure ok	Not Used
121	23	0.3922	0	Economizer damper position	Note 1	Economizer damper position %	Economizer damper position %
123	24	1	0		Note 1	SPECIAL	
123	24	1	0	Economizer Status	Note 1	State bit 0	Not Used
124	24	1	0		Note 1	State bit 1	Not Used
125	24	1	0		Note 1	State bit 2	Not Used
126	24	1	0		Note 1	Not Used	Not Used
127	24	1	0		Note 1	Exhaust Fan on	Not Used
128	24	1	0		Note 1	Outdoor Air suitable	Not Used
129	24	1	0		Note 1	Global Mode	Not Used
130	24	1	0		Note 1	Override mode	Not Used
131	25	1	0	System Status 1	Note 1	Read/Write to Eeprom	Read/Write to Eeprom
132	25	1	0		Note 1	Execute main control algorithm	Execute main control algorithm
133	25	1	0		Note 1	Transfer ECTO ROM default to factory ECTO	Transfer ECTO ROM default to factory ECTO
134	25	1	0		Note 1	Expansion bus error trapped	Not Used
135	25	1	0		Note 1	Expansion bus reset	Not Used
136	25	1	0		Note 1	No run mode	No run mode
137	25	1	0		Note 1	Expansion bus initialization	Not Used
138	25	1	0		Note 1	Startup	Startup
139	26	1	0	System Status 2	Note 1	Single phase mode	Not Used
140	26	1	0		Note 1	Local Access	Not Used
141	26	1	0		Note 1	System error trapped	System error trapped
142	26	1	0		Note 1	Save trapped errors to eeprom	Save trapped errors to eeprom
143	26	1	0		Note 1	Get system params from eeprom	Get system params from eeprom
144	26	1	0		Note 1	Cooling Mode	Cooling Mode
145	26	1	0		Note 1	Night Setback initialize	Reserved for warmup/cooldown
146	26	1	0		Note 1	Warmup mode enabled	Reserved for warmup/cooldown
147	27	1	0	Current System Operational Mode	Note 1	Current System Operational Mode	Current System Operational Mode
148	28	1	0	Device ID (Lsb)	Note 1	Reserved	Reserved
156	29	1	0	Device ID	Note 1	Device ID MSB	Device ID MSB
157	30	1	0	System Error Data	Note 1	Current Error Code (Unreliable)	Not Used
158	31	1	0	Stage Timer Status	Note 1	Stage Up enabled	Stage Up enabled
159	31	1	0		Note 1	Stage 2 on timer timeout	Stage 2 on timer timeout
160	31	1	0		Note 1	Stage 3 on timer timeout	Stage 3 on timer timeout
161	31	1	0		Note 1	Stage 4 on timer timeout	Stage 4 on timer timeout
162	31	1	0		Note 1	Stage down enabled	Stage down enabled
163	31	1	0		Note 1	Stage 2 off timer active	Stage 2 off timer active
164	31	1	0		Note 1	Stage 3 off timer active	Stage 3 off timer active

2ndry Array Offset[6]	Primary Array Offset	Scaling Multiplier	Scaling Constant	Data Category	Data filled in by other responses	Data filled in by 0x0605 with 0x30<=D29<=0x5f	Data filled in by 0x0605 with D29=0x60
165	31	1	0		Note 1	Stage 4 off timer active	Stage 4 off timer active
166	32	1	0	Sbus Demand Status	Note 1	G demand	G demand
167	32	1	0		Note 1	W1 demand	W1 demand
168	32	1	0		Note 1	W2 demand	W2 demand
169	32	1	0		Note 1	Y1 demand	Y1 demand
170	32	1	0		Note 1	Y2 demand	Y2 demand
171	32	1	0		Note 1	Smoke detected	Smoke detected
172	32	1	0		Note 1	Space Occupied	Space Occupied
173	32	1	0		Note 1	Room sensor remote control enabled	Room sensor remote control enabled
174	33	1	0	Expansion Bus Fail Status	Note 1	Board 1 failed	Not Used
175	33	1	0		Note 1	Board 2 failed	Not Used
176	33	1	0		Note 1	Board 3 failed	Not Used
177	33	1	0		Note 1	Board 4 failed	Not Used
178	33	1	0		Note 1	Board 5 failed	Not Used
179	33	1	0		Note 1	Board 6 failed	Not Used
180	33	1	0		Note 1	Board 7 failed	Not Used
181	33	1	0		Note 1	Board 8 failed	Not Used
182	34	1	0	System Status 7	Note 1	FAT on. (Fresh Air Temp)	Not Used
183	34	1	0		Note 1	Free cooling lockout	Not Used
184	34	1	0		Note 1	Not Used	Not Used
185	34	1	0		Note 1	Zone sensor demand holdoff in operation	Zone sensor demand holdoff in operation
186	34	1	0		Note 1	ECTO value signifies	Not Used
187	34	1	0		Note 1	Sbus has commanded blower on	Sbus has commanded blower on
188	34	1	0		Note 1	Not Used	Not Used
189	34	1	0		Note 1	Sbus has commanded economizer into override mode	Not Used
190	35	0.3922	0	Relative Humidity Sensor	Note 1	Relative Humidity Sensor	Relative Humidity Sensor
191	36	0.3922	0	Relative Humidity Sensor Setpoint	Note 1	Relative Humidity Sensor Setpoint	Relative Humidity Sensor Setpoint
192	37	1	0	RH1 Data	Note 1	Bypass solenoid 1 energized	Not Used
193	37	1	0		Note 1	Bypass solenoid 2 energized	Not Used
194	37	1	0		Note 1	Digital input 2 on	Not Used
195	37	1	0		Note 1	Not Used	Not Used
196	37	1	0		Note 1	Digital input 1 on	Not Used
197	37	1	0		Note 1	Not Used	Not Used
198	37	1	0		Note 1	Not Used	Not Used
199	37	1	0		Note 1	Not Used	Not Used
200	38	1	0	Reheat Status	Note 1	Not Used	Reserved for reheat control
201	38	1	0		Note 1	Not Used	
202	38	1	0		Note 1	Humiditrol option: Do not require blower on & occupied status …	
203	38	1	0		Note 1	Humiditrol option: Allow latent cooling demand to force …	
204	38	1	0		Note 1	Not Used	
205	38	1	0		Note 1	Not Used	
206	38	1	0		Note 1	Supermarket reheat on	
207	38	1	0		Note 1	Humiditrol reheat on	
208	39	1	0		Note 1	Not Used	Not Used
216	40	1	0		Note 1	Not Used	Not Used
224	41	1	0		Note 1	Not Used	Not Used
232	42	1	0		Note 1	Not Used	Not Used

2ndry Array Offset[6]	Primary Array Offset	Scaling Multiplier	Scaling Constant	Data Category	Data filled in by other responses	Data filled in by 0x0605 with 0x30<=D29<=0x5f	Data filled in by 0x0605 with D29=0x60
240	42	1	0		Note 1	Not Used	Not Used
248	44	1	0		Note 1	Not Used	Not Used
256	45	1	0		Note 1	Not Used	Not Used
264	46	1	0		Note 1	Not Used	Not Used
272	47	1	0		Note 1	Not Used	Not Used
280	48	1	0		Note 1	Not Used	Not Used
288	49	1	0		Note 1	Not Used	Not Used
296	50	1	0		Note 1	Not Used	Not Used
304	51	1	0		Note 1	Not Used	Not Used
312	52	1	0		Note 1	Not Used	Not Used
320	53	1	0		Note 1	Not Used	Not Used
328	54	1	0		Note 1	Not Used	Not Used
336	55	1	0		Note 1	Not Used	Not Used
344	56	1	0		Note 1	Not Used	Not Used
352	57	1	0		Note 1	Not Used	Not Used
360	58	1	0		Note 1	Not Used	Not Used
361	59	1	0	Number of Zones	Note 1	Number of Zones	Number of Zones
362	60	1	0	Economizer damper maximum travel	Note 1	Economizer damper maximum travel	Reserved for future economizer control
363	61	1	0		Note 1	Economizer direct damper control position	Reserved for future economizer control
364	62	1	0		Note 1	Economizer damper minimum position	Reserved for future economizer control
365	63	8.03E-05	0		Note 1	Economizer Enthalpy Setpoint	Reserved for future economizer control
366	64	8.03E-05	0		Note 1	Economizer Indoor Enthalpy	Reserved for future economizer control
367	65	8.03E-05	0		Note 1	Economizer Outdoor Enthalpy	Reserved for future economizer control
368	66	1	0		Note 1	Current Error Pointer	Current Error Pointer
369	67	8	0		Note 1	No Run Timeout Meter	No Run Timeout Meter
370	68	1	0		Note 1	No Run Error	No Run Error
371	69	128	0		Note 1	Night Setback Override Timer	Night Setback Override Timer
372	70	1	0	System Status 6	Note 1	Not Used	Not Used
373	70	1	0		Note 1	Not Used	Not Used
374	70	1	0		Note 1	Global Economizer Mode	Not Used
375	70	1	0		Note 1	Reheat Mode	Not Used
376	70	1	0		Note 1	Not Used	Not Used
377	70	1	0		Note 1	Not Used	Not Used
378	70	1	0		Note 1	Not Used	Not Used
379	70	1	0		Note 1	Not Used	Not Used
380	71	-0.25	100	Room Air Temp	Note 1	Room Air Temp	Room Air Temp
381	72	-0.25	100	°F	Occupied Heating Setpoint	Note 2	Note 2
382	73	-0.25	100	°F	Unoccupied Heating Setpoint	Note 2	Note 2
383	74	-0.25	100	°F	Occupied Cooling Setpoint	Note 2	Note 2
384	75	-0.25	100	°F	Occupied Cooling Setpoint	Note 2	Note 2
385	76	1	0	Compressor Configuration	Compressor #1 called on Y2	Note 2	Note 2
386	76	1	0		Compressor #2 called on Y2	Note 2	Note 2
387	76	1	0		Compressor #3 called on Y2	Note 2	Note 2
388	76	1	0		Compressor #4 called on Y2	Note 2	Note 2
389	76	1	0		Compressor #1 called on Y1	Note 2	Note 2
390	76	1	0		Compressor #2 called on Y1	Note 2	Note 2
391	76	1	0		Compressor #3 called on Y1	Note 2	Note 2
392	76	1	0		Compressor #4 called on Y1	Note 2	Note 2
393	77	1	0		Number of outdoor fans	Note 2	Note 2
394	78	1	0	Equipment Status #1 (IMC ver < 3.00)	Smoke control - damper open	Note 2	Note 2
395	78	1	0		Smoke control - exhaust fan on	Note 2	Note 2
396	78	1	0		Compressor 2 and 3 on Y2 only	Note 2	Note 2
397	78	1	0			Note 2	Note 2

2ndry Array Offset[6]	Primary Array Offset	Scaling Multiplier	Scaling Constant	Data Category	Data filled in by other responses	Data filled in by 0x0605 with 0x30<=D29<=0x5f	Data filled in by 0x0605 with D29=0x60
398	78	1	0			Note 2	Note 2
399	78	1	0			Note 2	Note 2
400	78	1	0			Note 2	Note 2
401	78	1	0			Note 2	Note 2
402	78	1	0	Equipment Status #1 (IMC ver >= 3.00)	Smoke control - damper open	Note 2	Note 2
403	78	1	0		Smoke control - exhaust fan on	Note 2	Note 2
404	78	1	0		Compressor 2 and 3 on Y2 only	Note 2	Note 2
405	78	1	0		Thermostat staging option	Note 2	Note 2
406	78	1	0		Thermostat staging option	Note 2	Note 2
407	78	1	0			Note 2	Note 2
408	78	1	0		Continuous blower in room sensor mode	Note 2	Note 2
409	78	1	0			Note 2	Note 2
410	80	1	0	Equipment Status #2	Not Used	Note 2	Note 2
411	80	1	0		Supplemental heat during defrost	Note 2	Note 2
412	80	1	0		defrost cycle time (lsb)	Note 2	Note 2
413	80	1	0		defrost cycle time (msb)	Note 2	Note 2
414	80	1	0		max defrost time (lsb)	Note 2	Note 2
415	80	1	0		max defrost time (msb)	Note 2	Note 2
416	80	1	0		lockout suplemental heat during heat pump warmup	Note 2	Note 2
417	80	1	0			Note 2	Note 2
418	81	1	0	Equipment Status #5	Compressor #1 called on room sensor stage 2	Note 2	Note 2
419	81	1	0		Compressor #2 called on room sensor stage 2	Note 2	Note 2
420	81	1	0		Compressor #3 called on room sensor stage 2	Note 2	Note 2
421	81	1	0		Compressor #4 called on room sensor stage 2	Note 2	Note 2
422	81	1	0		Compressor #1 called on room sensor stage 1	Note 2	Note 2
423	81	1	0		compressor #2 called on room sensor stage 1	Note 2	Note 2
424	81	1	0		Compressor #3 called on room sensor stage 1	Note 2	Note 2
425	81	1	0		Compressor #4 called on room sensor stage 1	Note 2	Note 2
426	82	1	0	Equipment Status #6	Compressor #1 called on room sensor stage 4	Note 2	Note 2
427	82	1	0		Compressor #2 called on room sensor stage 4	Note 2	Note 2
428	82	1	0		Compressor #3 called on room sensor stage 4	Note 2	Note 2
429	82	1	0		Compressor #4 called on room sensor stage 4	Note 2	Note 2
430	82	1	0		Compressor #1 called on room sensor stage 3	Note 2	Note 2
431	82	1	0		Compressor #2 called on room sensor stage 3	Note 2	Note 2
432	82	1	0		Compressor #3 called on room sensor stage 3	Note 2	Note 2
433	82	1	0		Compressor #4 called on room sensor stage 3	Note 2	Note 2
434	83	1	0	Reheat Mode	Reheat Mode	Note 2	Note 2
435	84	1	0	Expanded Device ID's	Expanded Device ID #1	Note 2	Note 2
436	85	1	0		Expanded Device ID #2	Note 2	Note 2
437	86	1	0		Expanded Device ID #3	Note 2	Note 2
438	87	1	0		Expanded Device ID #4	Note 2	Note 2
439	88	1	0		Expanded Device ID #5	Note 2	Note 2
440	89	1	0		Expanded Device ID #6	Note 2	Note 2
441	90	1	0		Expanded Device ID #7	Note 2	Note 2

2ndry Array Offset[6]	Primary Array Offset	Scaling Multiplier	Scaling Constant	Data Category	Data filled in by other responses	Data filled in by 0x0605 with 0x30<=D29<=0x5f	Data filled in by 0x0605 with D29=0x60
442	91	1	0		Expanded Device ID #8	Note 2	Note 2
443	92	1	0		Expanded Device ID #9	Note 2	Note 2
444	93	1	0		Expanded Device ID #10	Note 2	Note 2
445	94	1	0	Group 0x08 responses	Current System Mode	Note 2	Note 2
446	95	1	0	Error Block 1	Byte 0	Note 2	Note 2
447	96	1	0		Byte 1	Note 2	Note 2
448	97	1	0		Byte 2	Note 2	Note 2
449	98	1	0		Byte 3	Note 2	Note 2
450	99	1	0		Byte 4	Note 2	Note 2
451	100	1	0		Byte 5	Note 2	Note 2
452	101	1	0		Byte 6	Note 2	Note 2
453	102	1	0		Byte 7	Note 2	Note 2
454	103	1	0		Byte 8	Note 2	Note 2
455	104	1	0		Byte 9	Note 2	Note 2
456	105	1	0		Byte 10	Note 2	Note 2
457	106	1	0		Byte 11	Note 2	Note 2
458	107	1	0		Byte 12	Note 2	Note 2
459	108	1	0		Byte 13	Note 2	Note 2
460	109	1	0		Byte 14	Note 2	Note 2
461	110	1	0		Byte 15	Note 2	Note 2
462	111	1	0		Byte 16	Note 2	Note 2
463	112	1	0		Byte 17	Note 2	Note 2
464	113	1	0		Byte 18	Note 2	Note 2
465	114	1	0		Byte 19	Note 2	Note 2
466	115	1	0		Byte 20	Note 2	Note 2
467	116	1	0		Byte 21	Note 2	Note 2
468	117	1	0		Byte 22	Note 2	Note 2
469	118	1	0		Byte 23	Note 2	Note 2
470	119	1	0		Byte 24	Note 2	Note 2
471	120	1	0		Byte 25	Note 2	Note 2
472	121	1	0		Byte 26	Note 2	Note 2
473	122	1	0		Byte 27	Note 2	Note 2
474	123	1	0		Byte 28	Note 2	Note 2
475	124	1	0	Error Block 2	Byte 0	Note 2	Note 2
476	125	1	0		Byte 1	Note 2	Note 2
477	126	1	0		Byte 2	Note 2	Note 2
478	127	1	0		Byte 3	Note 2	Note 2
480	128	1	0		Byte 4	Note 2	Note 2
481	129	1	0		Byte 5	Note 2	Note 2
482	130	1	0		Byte 6	Note 2	Note 2
483	131	1	0		Byte 7	Note 2	Note 2
484	132	1	0		Byte 8	Note 2	Note 2
485	133	1	0		Byte 9	Note 2	Note 2
486	134	1	0		Byte 10	Note 2	Note 2
487	135	1	0		Byte 11	Note 2	Note 2
488	136	1	0		Byte 12	Note 2	Note 2
489	137	1	0		Byte 13	Note 2	Note 2
490	138	1	0		Byte 14	Note 2	Note 2
491	139	1	0		Byte 15	Note 2	Note 2
492	140	1	0		Byte 16	Note 2	Note 2
493	141	1	0		Byte 17	Note 2	Note 2
494	142	1	0		Byte 18	Note 2	Note 2
495	143	1	0		Byte 19	Note 2	Note 2
496	144	1	0		Byte 20	Note 2	Note 2
497	145	1	0		Byte 21	Note 2	Note 2
498	146	1	0		Byte 22	Note 2	Note 2
499	147	1	0		Byte 23	Note 2	Note 2
500	148	1	0		Byte 24	Note 2	Note 2
501	149	1	0		Byte 25	Note 2	Note 2
502	150	1	0		Byte 26	Note 2	Note 2
503	151	1	0		Byte 27	Note 2	Note 2
504	152	1	0		Byte 28	Note 2	Note 2

2ndry Array Offset[6]	Primary Array Offset	Scaling Multiplier	Scaling Constant	Data Category	Data filled in by other responses	Data filled in by 0x0605 with 0x30<=D29<=0x5f	Data filled in by 0x0605 with D29=0x60
505	153	1	0	Error Block 3	Byte 0	Note 2	Note 2
506	154	1	0		Byte 1	Note 2	Note 2
507	155	1	0		Byte 2	Note 2	Note 2
508	156	1	0		Byte 3	Note 2	Note 2
509	157	1	0		Byte 4	Note 2	Note 2
510	158	1	0		Byte 5	Note 2	Note 2
511	159	1	0		Byte 6	Note 2	Note 2
512	160	1	0		Byte 7	Note 2	Note 2
513	161	1	0		Byte 8	Note 2	Note 2
514	162	1	0		Byte 9	Note 2	Note 2
515	163	1	0		Byte 10	Note 2	Note 2
516	164	1	0		Byte 11	Note 2	Note 2
517	165	1	0		Byte 12	Note 2	Note 2
518	166	1	0		Byte 13	Note 2	Note 2
519	167	1	0		Byte 14	Note 2	Note 2
520	168	1	0		Byte 15	Note 2	Note 2
521	169	1	0		Byte 16	Note 2	Note 2
522	170	1	0		Byte 17	Note 2	Note 2
523	171	1	0		Byte 18	Note 2	Note 2
524	172	1	0		Byte 19	Note 2	Note 2
525	173	1	0		Byte 20	Note 2	Note 2
526	174	1	0		Byte 21	Note 2	Note 2
527	175	1	0		Byte 22	Note 2	Note 2
528	176	1	0		Byte 23	Note 2	Note 2
529	177	1	0		Byte 24	Note 2	Note 2
530	178	1	0		Byte 25	Note 2	Note 2
531	179	1	0		Byte 26	Note 2	Note 2
532	180	1	0		Byte 27	Note 2	Note 2
533	181	1	0		Byte 28	Note 2	Note 2
534	182	1	0	Error Buffer Pointer	Error Buffer Pointer	Note 2	Note 2
535	183	1	0	Version Number	Version Number Byte 1	Note 2	Note 2
536	184	1	0		Version Number Byte 2	Note 2	Note 2
537	185	1	0		Version Number Byte 3	Note 2	Note 2
538	186	1	0		Version Number Byte 4	Note 2	Note 2
539	187	1	0		Version Number Byte 5	Note 2	Note 2
540	188	1	0		Version Number Byte 6	Note 2	Note 2
541	189	1	0		Version Number Byte 7	Note 2	Note 2
542	190	1	0		Version Number Byte 8	Note 2	Note 2
543	191	1	0	Cooling / heating Off Delays	Note 1	Note 2	Y1 Off Delay
544	191	1	0		Note 1	Note 2	Y2 Off Delay
545	191	1	0		Note 1	Note 2	Y3 Off Delay
546	191	1	0		Note 1	Note 2	Y4 Off Delay
547	191	1	0		Note 1	Note 2	Note 3
548	191	1	0		Note 1	Note 2	Note 3
549	191	1	0		Note 1	Note 2	Note 3
550	191	1	0		Note 1	Note 2	Note 3
551...571	192…212						Reserved
572	213…244	1	0	Device copyrights - Application software copyright	Bytes 0...31 of command 0x0267 when 44 bytes of data are returned.	Note 2	Note 2
572	245…255	1	0	Device copyrights - Application software compile date	Bytes 32...43 of command 0x0267 when 44 bytes of data are returned.	Note 2	Note 2
572	257…288	1	0	Device copyrights - Application software copyright	Bytes 0...31 of command 0x0267 when 88 bytes of data are returned.		
572	289…344	1	0	Device copyrights - Application software compile date	Bytes 32... 43 of command 0x0267 when 88 bytes of data are returned.	Note 2	Note 2

2ndry Array Offset[6]	Primary Array Offset	Scaling Multiplier	Scaling Constant	Data Category	Data filled in by other responses	Data filled in by 0x0605 with 0x30<=D29<=0x5f	Data filled in by 0x0605 with D29=0x60
572	257…288	1	0	Device copyrights - Application software copyright	Bytes 44... 75 of command 0x0267 when 88 bytes of data are returned.	Note 2	Note 2
572	289…344	1	0	Device copyrights - Application software compile date	Bytes 76... 87 of command 0x0267 when 88 bytes of data are returned.	Note 2	Note 2
573	346	1	0	Gateway Status Data	D00: Unit Status Output	Note 2	Note 2
573	347	-.25	100	Gateway Status Data	D01: Effective Space Temperature Output	Note 2	Note 2
573	348	-0.6729	164.45	Gateway Status Data	D02: Discharge Air Temperature Output	Note 2	Note 2
573	349	1	0	Gateway Status Data	D03: Effective Occupancy Output	Note 2	Note 2
573	350	-0.6360	131.56	Gateway Status Data	D04: Local Outdoor Air Temperature Output	Note 2	Note 2
573	351	-0.25	100	Gateway Status Data	D05: Local Space Temperature Output	Note 2	Note 2
573	352	1	0	Gateway Status Data	D06: Outdoor Air Damper Output	Note 2	Note 2
573	353	1	0	Gateway Status Data	D07: Primary Heating Output	Note 2	Note 2
573	354	1	0	Gateway Status Data	D08: Secondary Heat Output	Note 2	Note 2
573	355	1	0	Gateway Status Data	D09: Primary Cooling Output	Note 2	Note 2
573	356	1	0	Gateway Status Data	D10: Economizer Enabled Output	Note 2	Note 2
573	357	1	0	Gateway Status Data	D11: Supply Fan Status	Note 2	Note 2
573	358	-0.25	100	Gateway Status Data	D12: Effective Setpoint Output	Note 2	Note 2
573	359	1	0	Gateway Status Data	D13: Currently displayed error code	Note 2	Note 2
573	360	1	0	Gateway Status Data	D14: Point to next available error storage location.	Note 2	Note 2
573	361	1	0	Gateway Status Data	D15: Most recent error code	Note 2	Note 2
573	362	1	0	Gateway Status Data	D16: 2nd most recent error code	Note 2	Note 2
573	363	1	0	Gateway Status Data	D17: 3rd most recent error code	Note 2	Note 2
573	364	1	0	Gateway Status Data	D18: 4th most recent error code	Note 2	Note 2
573	365	1	0	Gateway Status Data	D19: 5th most recent error code	Note 2	Note 2
573	366	1	0	Gateway Status Data	D20: 6th most recent error code	Note 2	Note 2
573	367	1	0	Gateway Status Data	D21: 7th most recent error code	Note 2	Note 2
573	368	1	0	Gateway Status Data	D22: 8th most recent error code	Note 2	Note 2
573	369	1	0	Gateway Status Data	D23: 9th most recent error code	Note 2	Note 2
573	370	1	0	Gateway Status Data	D24: 10th most recent error code	Note 2	Note 2
573	371	1	0	Gateway Status Data	D25: A133 (GP) digital inputs (2) and output (1)	Note 2	Note 2
573	372	0.03992	0	Gateway Status Data	D26: A133 (GP) analog input channel 1	Note 2	Note 2
573	373	0.03992	0	Gateway Status Data	D27: A133 (GP) analog input channel 2	Note 2	Note 2
573	374	0.03992	0	Gateway Status Data	D28: A133 (GP) analog input channel 3	Note 2	Note 2
573	375	0.03992	0	Gateway Status Data	D29: A133 (GP) analog input channel 4	Note 2	Note 2
573	376	0.1	0	Gateway Status Data	D30: A133 (GP) analog output channel 1	Note 2	Note 2
573	377	0.1	0	Gateway Status Data	D31: A133 (GP) analog output channel 2	Note 2	Note 2
573	378	0.1	0	Gateway Status Data	D32: A133 (GP) analog output ch1 setpoint	Note 2	Note 2
573	379	0.1	0	Gateway Status Data	D33: A133 (GP) analog output ch2 setpoint	Note 2	Note 2
573	380	7.843	0	Gateway Status Data	D34: Space CO2 Sensor Output (effective)	Note 2	Note 2
573	381	7.843	0	Gateway Status Data	D35: Space CO2 Sensor Output (local)	Note 2	Note 2
573	382	0.39216	0	Gateway Status Data	D36: Space Humidity Output (effective)	Note 2	Note 2
573	383	0.39216	0	Gateway Status Data	D37: Space Humidity Output (local)	Note 2	Note 2
573	384	0.39216	0	Gateway Status Data	D38: Effective Space Dehumidification Setpoint Output	Note 2	Note 2

2ndry Array Offset[6]	Primary Array Offset	Scaling Multiplier	Scaling Constant	Data Category	Data filled in by other responses	Data filled in by 0x0605 with 0x30<=D29<=0x5f	Data filled in by 0x0605 with D29=0x60
573	385	1	0	Gateway Status Data	D39: Dehumidification Status Output	Note 2	Note 2
573	386	-0.6792	164.45	Gateway Status Data	D40: Return Air Temperature Output	Note 2	Note 2
573	387	0.0039216	-0.5	Gateway Status Data	D41: Building Static Pressure	Note 2	Note 2
573	388	0.01961	0	Gateway Status Data	D42: Duct Static Pressure	Note 2	Note 2
573	389	0.01961	0	Gateway Status Data	D43: Effective Duct Static Pressure Setpoint Output	Note 2	Note 2
573	390	1	0	Gateway Status Data	D44: Exhaust Fan Capacity Output	Note 2	Note 2
573	391	-0.6792	164.45	Gateway Status Data	D45: Effective Discharge Air Temp Setpoint Output	Note 2	Note 2
573	392	1		Gateway Status Data	D46: A133 (VAV) digital inputs (2) and output (1)	Note 2	Note 2
573	393	0.039216	0	Gateway Status Data	D47: A133 (VAV) analog input channel 1	Note 2	Note 2
573	394	0.039216	0	Gateway Status Data	D48: A133 (VAV) analog input channel 2	Note 2	Note 2
573	395	0.039216	0	Gateway Status Data	D49: A133 (VAV) analog input channel 3	Note 2	Note 2
573	396	0.039216	0	Gateway Status Data	D50: A133 (VAV) analog input channel 4	Note 2	Note 2
573	397	1	0	Gateway Status Data	D51: A133 (MGV) digital inputs (2) and output (1)	Note 2	Note 2
573	398	0.039216	0	Gateway Status Data	D52: A133 (MGV) analog input channel 1	Note 2	Note 2
573	399	0.039216	0	Gateway Status Data	D53: A133 (MGV) analog input channel 2	Note 2	Note 2
573	400	0.039216	0	Gateway Status Data	D54: A133 (MGV) analog input channel 3	Note 2	Note 2
573	401	0.039216	0	Gateway Status Data	D55: A133 (MGV) analog input channel 4	Note 2	Note 2
574	402	1	0		// Special: Used to store version info from 0x06 0x04		

7.3 Map Descriptor Examples

7.3.1 Specifying the Secondary Data Array

This example is almost identical to the Client Map Descriptor section's first example. In this case we have also defined the secondary storage array. Note that this driver uses the offset for both arrays.

```
//    Client Side Map Descriptors
Map Descriptors
Map_Descriptor_Name  , Data_Array_Offset  , Data_Array_Name  , DA_Bit_Name     , Function  , Node_Name  , Length  , Scan_Interval  , Lennox_Func
MD_Read_All_01       , 0                  , DA_N1_Data       , DA_N1_BitData  , RDBC      , Node_1     , 950     , 1.0s           , Read Everything
```

In the above example:

- Data_Array_Name – This is the primary Data Array.
- DA_Bit_Name – This is the secondary Data Array. Don't be misled by the parameter name "DA_Bit_Name" the Data Array can have any data format (see data array section).

7.3.2 Bus Arbitration

This Map Descriptor is required if you want you Server to allow technician/guest devices to take control of the bus. It sends a special message periodically which these devices listen for and to which they respond telling the Server to go offline, if required. **Section 7.5** has more information on bus arbitration.

```
//    Client Side Map Descriptors
Map Descriptors
Map_Descriptor_Name  , Data_Array_Offset  , Data_Array_Name  , Function  , Node_Name  , Length  , Scan_Interval  , Lennox_Func
MD_Read_All_01       , 0                  , DA_N1_Data       , RDBC      , Node_DE    , 950     , 5.0s           , Request Device ID
```

In the above example:

- Node_Name – The address of this node must be 0xdf (equal to 223 decimal).
- Scan_Interval – This message doesn't need to be sent too often.
- Lennox_Func – The 'guest' device can:
  - Respond to this message with the device ID. This has no effect on the bus master
  - Respond with a command to put the bus master (this connection) offline
  - Remain silent (and invisible). The driver has been programmed so that the timeouts that may arise in this situation will not generate timeout errors

7.4 Scaling Considerations

Many drivers allow users to specify scaling with each Map Descriptor. This driver ignores such scaling since many Map Descriptors poll for composite data sets and the scaling associated with a Map Descriptor would have to be applied to all the data items.

The driver does the following automatic scaling:

- When the driver sends data (writes) to a Lennox device it scales the data using the scaling information provided in **Section 7.1.1**.
- When the driver responds to a poll, it does not use scaling. The values extracted from the Data Arrays are sent 'as is' in the messages.
- When a Lennox Client receives a response from a Lennox device it scales the data using the scaling information in **Section 7.1.1** before storing the data.

7.5 Guest Devices, Bus Arbitration (Putting the Bus Master/Nodes Offline)

The following notes are reprinted from the Lennox Protocol Specification:

When a "Request Device ID' poll is sent to address 0xDF = 223, the remote device has an opportunity to:

- Give no response – The technician device may be eavesdropping on the network and not want to make its presence known.
- Respond with the device ID
- Respond with a command that tells the message sender (the FieldServer) to go offline. It does so when the technician device wants to take control of the network for is own purposes. If the technician device has taken control then it can relinquish control explicitly by sending a message to the FieldServer to go back online or should the connection go silent for 5 minutes then the FieldServer can reclaim control.
- When the FieldServer Server is offline it stops polling remote devices.
- What if the FieldServer is configured as a Server, emulating a Lennox device? It's possible that the master on such a network can tell the FieldServer node to go offline. IN such cases the FieldServer would stop responding to polls from other devices on the network until the same master that put the FieldServer node offline, puts it back online. Again, if the FieldServer node sees no messages for 5 minutes it will put itself back online.

7.6 Setting the FieldServer Node_ID

All Lennox driver messages contain a source and destination station Node_ID. For this reason, it is important to set the Node_ID of the FieldServer itself. The following fragment should be modified and included in the configuration CSV file.

```
FieldServer
Title                                , System_Node_Id
Lennox Bus Master                    , 255
```

Note:

Lennox requires that nodes such as the FieldServer are allocated addresses in the range 0xE1-0xEF.

The following address range is used for 'protocol translators':

Hex	Dec
0xe1	225
0xe2	226
0xe3	227
0xe4	228
0xe5	229
0xe6	230
0xe7	231
0xe8	232
0xe9	233
0xea	234
0xeb	235
0xec	236
0xed	237
0xee	238
0xef	239

7.7 Alarm Table Processing

The driver automatically reads, sorts and stores the alarm table whenever a new alarm is detected.

When the response from one of 'Get Error Buffer Pointer' or 'Send Expanded global variables status' is processed the driver checks the value of the Error Buffer Pointer (EBP) before writing it to the Data Array. If the value is different, this is considered to be the indication that a new alarm has occurred.

If the EBP is different the driver automatically kicks off three polls, one to read each Error Block. When the response from the 3rd poll is received, the driver prepares a sorted table and searches for a passive Map Descriptor with a special keyword. Once found, the sorted table is stored along with three extra values. If a storage location has not been defined, Error Message #38 will be returned. If the EIDX value is not between 0 and 83 then the alarms cannot be sorted and the data is discarded. Error message #39 is returned in this case.

87 values are written to the Data Array. The 1st 84 values represent the sorted alarm table. The 85th value is the value of the EBP; the 86th value is the value of the EIDX and the 87th value reports whether the buffer has overflowed. If more than 84 alarms have been received since the alarms were last cleared then the table has overflowed and some alarms will have been overwritten. This is indicated with a value of 1. A value of zero indicates that there is no overflow.

It is the responsibility of the operator to clear the alarm buffer. We recommend that you animate an alarm with the value of the overflow register so that control room personnel can see when the alarm buffer has overflowed. We further recommend that you animate an alarm based on the out of range value for the EIDX to alert the control room personnel when the alarm table is not valid. We also recommend that you follow the alarm clearing strategy discussed in one of the examples below.

7.8 Map Descriptor Examples

7.8.1 Information required for Sorted Alarm Table Storage

```
Map Descriptors
Map_Descriptor_Name  , Data_Array_Offset  , Data_Array_Name     , Function  , Node_Name  , Length  , Lennox_Func
MDP_01               , 0                 , DA_ALARM_TABLE     , Server    , NODE01     , 87      , Sorted Alarm List
```

In the above example:

- Data_Array_Name – Data is stored in the Data Array specified here.
- Function – Must be 'Server' or 'Passive'.
- Lennox_Func – Keyword used to by driver to recognize this Map Descriptor.
- Length – 87 Elements are required to store the table (84 elements) and the three additional data items.

7.8.2 Clearing the Alarm Table

The following notes and Map Descriptor example outline a strategy for clearing the alarm table. Animate a push button on your MMI so that when it is pushed it sets/toggles the value of a single element of a Data Array in the FieldServer. Now create a wrbx (write on update) Map Descriptor in the FieldServer that monitors the same element in the same Data Array as in the HMI animation. Whenever the operator pushes the button the FieldServer has its value in the Data Array updated. Whenever the value is updated (even if the same value is written again) the Map Descriptor will be triggered.

```
Map Descriptors
Map_Descriptor_Name    , Data_Array_Offset  , Data_Array_Name   , Function  , Node_Name  , Length  , Lennox_Func
Write_97_Map Descriptor , 59                , DA_FLOAT1_OUT     , wrbx      , Node01     , 1       , Clear error buffer
```

In the above example:

- Data_Array_Name – When the value found at this offset in this Data Array is set the wrbx is triggered. You can use any DA and offset.
- Function – The function wrbx ensures that the table is only cleared when the Map Descriptor is triggered. If you use wrbc then the message will be sent periodically.
- Length – The length should be 1.
- Lennox_Func – This is the command to send a message to clear the Lennox device's error buffers.

7.9 Using Node_Type to Control ‘Read Everything’ Command

The sets of polls sent to read everything is controlled by the parameter ‘NODE_TYPE’ which is specified on the node. If “NODE_TYPE” is not specified in the CSV file, the default is “Full”

The applicability of the plc_types is dependent on the type of Lennox Device. If the device is a Lennox Protocol Gateway (SB Gateway) then the only valid selection is ‘gateway’. The other selections (gateway is excluded) may be used for other devices.

plc_type = 'Full'

Send current heating and cooling setpoints"
Send Reheat Status"
Send Equipment Configuration"
Request for device ID"
Request for expanded device ID"
Request for system mode"
Send Expanded global variables status"
Send error block 0"
Send error block 1"
Send error block 2"
Send error buffer pointer"
Send version number"

plc_type = 'Set1'

Send current heating and cooling setpoints"
Send Expanded global variables status"
Send error block 0"
Send error block 1"
Send error block 2"
Send error buffer pointer"
Send version number"

plc_type = 'Set2'

Send Expanded global variables status"
Send error block 0"
Send error block 1"
Send error block 2"
Send error buffer pointer"

plc_type = 'Minimal'

Send error block 0"
Send error block 1"
Send error block 2"
Send error buffer pointer"

plc_type = 'Gateway'

Send Gateway Global Status
IMC Version Number
Request for device ID

7.10 Addressing

When each controller powers up, it configures itself to its bus address by either reading the positions of address DIP switches on the board, by using firmware configured addresses or a combination of the two. Some devices, such as the Notifact, that can only be installed one to a network are factory assigned an address. The Sbus address space can accommodate 255 devices. In practice the number of devices is limited by the installed hardware and software. Address zero is reserved for global addressing of all devices on the bus. If address zero is chosen on an Sbus controller, it will not respond to communications until the address is changed and may issue an error message. By convention, the Sbus assigns types of controllers to particular address ranges, corresponding to the device type. The address range f slave type controllers is 0x01 – 0x7f (0 – 31 decimal). With the exception of zone modules, the address range of Server type controllers is 0x80 – 0xff (128 –255 decimal). Typically, the Server type controller address and device ID will match.

7.11 MultiDrop

If there is more than one FieldServer Server node on a single connection, then the Multidrop capability must be enabled. The following CSV file fragment illustrates how this can be done.

```
Connections
Port  Baud  Data_Bits  Stop_Bits  Parity  Protocol  Multidrop_Mode
R1    9600  8          1          None    Lennox    Enable
```

8 Driver Error Messages

Message	Description
Lnnx:#1 Err. Illegal Node_ID=%d. Permitted=1...255	The driver permits Node_ID's in the range 1 to 255. Zero cannot be used as this is the 'broadcast' Node_ID for the protocol. Normally Lennox devices can only be addressed in the range 1-31, but the driver allows a larger range of ID's to be used to facilitate communication with Technician and other specially devices.[7]
Lnnx:#2 Err. Duplicate node=%d on same port.	Two nodes may not have the same Node_ID if they are connected to the same port. For Server applications, if the nodes are not connected to a specific port you can only define one node for each Node_ID.[7]
Lnnx:#4 Err. Keyword / Cmd Id's invalid. Md=<%s>	Check the spelling and spacing of keywords. The driver is not case sensitive. If you specified the function using lennox_cmd0/1 parameters then values given were not recognized. Use **Section 7.1.1** to check your configuration. [7]
Lnnx:#5 Err. Length rqd. Min=%d Max=%d Md=%d	The length parameter has been incorrectly specified. The error message reports the min and max values permitted. Refer also to **Section 7.1.1**. [7]
Lnnx:#6 Err. Map Descriptor length required.	The required length parameter has not been specified. Refer to **Section 7.1.1**. [7]
Lnnx:#7 Err. No Func Keyword. Specify cmd0/1	The Map Descriptor was rejected because no Lennox_Func parameter was specified. You must specify the Lennox_Func or the Lennox_Cmd0/1.[7]
Lnnx:#8 Err. Unknown Function. cmd0=0x%02x cmd1=0x%02x keyword=<%s>	Either the keyword(s) specified for Lennox_Func parameter OR the values specified with the lennox_cmd0/1 parameters have not been recognized by the driver. Use **Section 7.1.1** to select a suitable keyword.[7]
Lnnx:#9 Err. Keyword unsuitable for read/write. Md=<%s>	The poll command (as specified with the Lennox_Func or Lennox_cmd0/1 parameters) is not suitable for a read or a write as it is a response or Ack message as defined by the protocol. Use **Section 7.1.1** to select a suitable command.[7]
Lnnx:#10 Err. Function mis-match. Read expected. Md=<%s>	The poll command (as specified with the Lennox_Func or Lennox_cmd0/1 parameters) is not suitable for a write. Either change the command or change the 'function' to *rdbc*, or *rdb*.[7]
Lnnx:#11 Err. Function mis-match. Write expected. Md=<%s>	The poll command specified with the Lennox_Func or Lennox_cmd0/1 parameters is not suitable for a read. Change the 'function' to *wrbc*, *wrb* or *wrbx*.[7]
Lnnx:#12 Err. Srvr rcvd an unknown cmd. 0x%02x:0x%02x	The Server received an unknown command. This could be a single corrupted message or because the firmware on a Lennox unit has implemented some commands that had not been defined when the driver was developed.[8]
Lnnx:#13 Err. Srvr rcvd a resp. Can't Parse. 0x%02x:0x%02x	The Server was expecting a poll but received a response. You cannot take any corrective action.[8]
Lnnx:#14 Err. Client recvd a broadcast response. Src=%d cmd0/1=0x%02x:0x%02x	The Client received a response addressed to the broadcast address (Node_ID=0). This driver cannot process broadcast messages. Disable the broadcast poll in the device which sent the message. The message indicates the number (in decimal) of the source Node_ID.
Lnnx:#15 Err. Client doesnt recognize command=0x%02x:0x%02x	This is similar to error #12 except that it applies to the Client receiving a response.

[7] Modify the configuration CSV file, download the modified file to the FieldServer and then reset the FieldServer
[8] If the error occurs frequently take a log (find instructions in the unit start-up guide) and report the problem to FieldServer support.

Message	Description
Lnnx:#16 FYI. Unit(=0x%02x) put this node(=0x%02x) offline.	A device has responded to a poll from the master, with a message telling the master to go offline. The master will stop polling until it is put back online or until the connection is quiet for a period of time.[9]
Lnnx:#17 Err. Client could not parse command=0x%02x:0x%02x	The difference between this message and error #15 is that in this case the message is legitimate (in terms of the protocol) but the driver has not implemented a method for analyzing the message.[10]
Lnnx:#18 Err. Array too short. Rqd/Act=%d/%d Md=<%s>	This message is printed when the Client is forming a write message and in doing so attempts to extract data from a Data Array but finds that the offset specified points beyond the end of the array. Review your configuration file using the information provided in the message.
Lnnx:#19 Err. Cant poll. Cmd code is unknown. 0x%02x 0x%02x	You should never see this message.[10]
Lnnx:#20a/b/c/d/e/f/g/h Err. Diagnostic #3. Call Support.	
Lnnx:#21 FYI. Unit(=0x%02x) put this node(=0x%02x) back online.	A device has taken control of the bus and is now relinquishing control back to the original bus master (in this case the FieldServer Client) When this happens the Client/master starts polling again.[9]
Lnnx:#22 FYI. Unit(=0x%02x) not allowed to put node(=0x%02x) back online because this Unit(=0x%02x) put the node offline.	When a device has taken control of the bus only that device can put the original bus master back online.[9]
Lnnx:#23 FYI. Unit(=0x%02x) put this node(=0x%02x) offline.	This message identifies the remote node that put this node offline. When a node is offline it cannot respond to polls from other devices.[9]
Lnnx:#24 Err. Version number is min of 4 bytes	When configured as a Server, this driver responds to a poll for the version number with a variable number of bytes. The number sent in the reply is controlled by the length of the Server Map Descriptor.
Lnnx:#25 Err. Version number is max of 8 bytes	
Lnnx:#26 FYI. Node(=0x%02x) timed out to online mode.	A node that was put offline has not received any message for the timeout period and is therefore putting itself back online.[9]
Lnnx:#27 FYI. Bus Mstr(=0x%02x) is reclaiming online status. Polling enabled.	A Client connection has reclaimed bus master status as there has been no activity on the connection for the timeout period.[9]
Lnnx:#28 FYI. Unit(=0x%02x) put this Bus Mstr(=0x%02x) back online.	The identified remote device explicitly relinquished control and puts the connection back online.[9]
Lnnx:#29a Err. Md=%s Array=%s too short. Rqd/Act=%d/%d	The Server is storing information sent by – a) Group HVAC command b) Clear Timers command c) Clear Error Buffer command. These commands send variable length messages. The error message tells you how long to make the Map Descriptor. Review your configuration file using the information provided in the message.[11]

[9] This message is informational and may be safely ignored. More information on Bus Arbitration is provided in **Section** 7.5.
[10] If the error occurs frequently take a log (find instructions in the unit start-up guide) and report the problem to FieldServer support.
[11] Modify the configuration CSV file, download the modified file to the FieldServer and then reset the FieldServer.

Message	Description
Lnnx:#30 Err. MD=<%s> is too short. Rqd=%d.	The Map Descriptor used to expose driver stats is too short. The message reports the required length. If not corrected, the driver will continue to operate correctly but will not expose operating stats. See section 72.[12]
lnnx:#31 FYI. You could have used a Map Descriptor called <%s> to expose diagnostic info.	Refer to **Section 9**.
Lnnx:#33 FYI. Bus Mstr(=0x%02x) is reclaiming online status. Polling enabled.	See error message #27.
Lnnx:#34 FYI. Bus Mstr(=0x%02x) is offline. Polling inhibited.	This message is printed each time an offline connection receives a poll from a device other than the device that put the connection offline.[13]
Lnnx:#35 Err. Server recvd broadcast msg. Cmd=0x%02x:0x%02x	The Server received a broadcast message which It is unable to process. See error message #14.
Lnnx:#36 Err. Duplicate Nodes on same port(=%d).	The driver has found a duplicate node on the same port. See error message #2.
Lnnx:#37a Err. Array=%s to short. Rqd/Act=%d/%d	Variation 37a is printed when data is being written to the primary array. Variation 37b is printed when data is being written to the secondary array. The error reports the actual length of the array vs. the length required. Note that the length required depends on the number of items being stored as well as the offset specified in the Map Descriptor.[12]
Lnnx:#38* Err. No place to store sorted alarms. Discarding	A passive Map Descriptor with the name “Sorted Alarm List” could not be found. Therefore, the driver could not store the sorted alarm list. You can ignore this message if you are not interested in storing the sorted alarm list, otherwise define this Map Descriptor. See **Section 7.7** for additional information. Multiple occurrences of this message are suppressed.
Lnnx:#39 FYI. Incorrect ebp %d, discarding err block info	You can ignore this message if you are not interested in storing the sorted alarm list. Only certain values of the Error Buffer Pointer (EBP) are valid. If the value is invalid then the driver cannot determine how to sort the table and is thus forced to discard the sorted alarm data.
Lnnx:#40 FYI. <%s> Not have enough room to store %d long ,discarding err block info	The number of elements required to store the sorted alarm table is 87. The ‘Length’ parameter specified on the Map Descriptor must be at least 87 and the Data Array must be at least 87 elements long.[12]
Lnnx:#41 FYI. Node=%s type not defined. Assuming full polling.	If you are using the ‘Read Everything’ method then the driver uses the Node_Type specified on the node in the csv field to determine which polls to include in the set of ‘everything’. More information is provided in **Section 7.7**. This message reports that the Node_Type has not been specified.[12]
Lnnx:#42 Err. Function mis-match. Server expected. Md=<%s>	The Map Descriptor used to store the ‘Sorted Alarm List must be passive (or Server). **Section 7.7** provides additional information.[12]

[12] Modify the configuration CSV file, download the modified file to the FieldServer and then reset the FieldServer.
[13] If the error occurs frequently take a log (find instructions in the unit start-up guide) and report the problem to FieldServer support.

Message	Description
Lnnx:#43a FYI. Node=%s Read Everything with full poll list.	These messages are provided for your information only and may be safely ignored if they confirm your expectations. The messages provide feedback (for each node) on the Node_Type specified in the csv file. The Node_Type determines which polls form the set of polls used to read everything.
Lnnx:#43b FYI. Node=%s Read Everything with reduced poll list.(1)	
Lnnx:#43c FYI. Node=%s Read Everything with reduced poll list.(2)	
Lnnx:#43d FYI. Node=%s Read Everything with minimal poll list.	
Lnnx:#44 Err. Map descriptor length has to be 44 or 88	If you configure a Map Descriptor to read the remote device's copyright information then the length of the Map Descriptor must be 44 or 88. If the remote device has a 'bootloader' then length must be 88, otherwise a length of 44 is sufficient.[14]
Lnnx:#45 Err. MD Length must be an even number and <= 6	If a Map Descriptor is configured to 'Set AD input to override value' then the length of the Map Descriptor must be 2, 4 or 6.[14]
Lnnx:#46 Err. Map descriptor length has to be 44 or 88	A server map descriptor has been configured to serve the device's copyright information. The length of the MD must be 44 or 88. 88 bytes indicates that the device has a boot loader.[14]
Lnnx:#47 Err. Local_Node_Id / System_Node_ID not set.	Messages sent to a Lennox device require two addresses – source and destination. Previously the driver used the System_Node_ID as the source address but in current versions the configuration requires that the user define a 'Local_Node_ID' on the node descriptor to be used as the source address.[14]
Lnnx:#48 Err. Polling Suppressed until config is corrected	See error message #47 which reports the cause.
"Lnnx:#49 FYI. Invalid System_Node_ID(%d) , Valid=225-239 incl.	To provide backward compatibility the driver still allows the System_Node_ID to be used as the source address for messages to the Lennox device. The range of valid addresses is limited.[14]
Lnnx:#50a/b FYI. Responding as station 225 (0xe1)	This message is printed by the Server side of the driver when a remote client polled for a device Id. The response is based on a DA value. If the value is zero and the server doesn't have a System_Node_ID set then the driver responds as station 225. You do not need to take any corrective action. If you wish you can edit the configuration and specify a System_Node_ID to change the address of the responding station.[14]
Lnnx:#51 FYI. Invalid Local_Node_Id(%d), Valid=225-239 incl.	This is a warning only. The message prints the range of recommended addresses. They are not mandatory, so if you are happy with the configuration you do not need to take any action.[14]
Lnnx:#52 FYI. Unit(=0x%02x) put this node(=0x%02x) online.	This message is printed each time a node that was offline is sent back to the online state. No corrective action is required.
Lnnx:#52 FYI. Commanded to go offline	A remote device has put this node offline. The message is printed for your information only. When a node is offline, it stops polling. No corrective action is required.

[14] Modify the configuration CSV file, download the modified file to the FieldServer and then reset the FieldServer.

Message	Description
Lnnx:#52c FYI. Unit(=0x%02x) attempts to put node(=0x%02x) online. Lnnx:#52c FYI. But the unit was put offline by node(=0x%02x) Or Lnnx:#52c FYI. But the unit was online already.	Only the station that put a node offline can put it back online. This message is printed if the 'go online' command came from a different station. Wait for the device to timeout back to an online state. This will happen if the connection is quiet for the timeout period. Alternatively try getting the original station that put the node offline to send the 'go online' command. A 'go online' command may be received when the station is already online. No corrective action is required.
Lnnx:#53 Err. On timeout Failed to get new Node_id.(timeout)	If the 'Lennox_On_Timeout_Behaviour' is set to 'poll_for_new_node_id'; when a timeout occurs, the driver sends a broadcast poll requesting the Lennox device ID. If there is no response to the poll, the driver notifies the kernel that there was a timeout. The driver prints this message once and then suppresses it unless the timeout occurs again. Refer to Client Side Node Descriptors section.
Lnnx:#54 Err. Lennox_On_timeout_Behaviour: Poll for new Node Id "Lnnx:#54 Err. Node_ID: New=Old=%d. Treating like timeout.	If the Node_ID in response to the broadcast poll described in Err #53 is unchanged it means the original timeout was valid and thus the driver notifies the kernel of the timeout. There is no corrective action you can take. Refer to Client Side Node Descriptors section.
Lnnx:#55 FYI Lennox Node_ID changed. From=%d To=%d	If the Node_ID in response to the broadcast poll described in Err #53 is changed, then the driver continues to poll using the new ID. Refer to Client Side Node Descriptors section.

9 Standard Operating Statistics

The driver reports statistics according to the FieldServer standards. The following notes describe some aspects of standard statistic reporting which are peculiar to this driver.

Operating Stat	Notes
Protocol Errors	Increments by 1 each time • The master fails to send a poll for a reason such as being unable to recognize the command. • The master can't process a response because it doesn't recognize the message as complete. This could arise if the message had the wrong header or the message length was invalid. • The master cannot parse a response. This could happen if the response header was invalid, the message was incorrectly formatted or the command codes were not recognized. • The Server receives a message with an invalid header. This probably indicates that the driver is receiving messages from some other protocol or that the connection settings (baud, parity etc. are incorrect.) • The Server receives a message which appears to be incomplete because the length was invalid. • The Server receives a message whose command codes are not recognized. • The Server receives a response when a poll was expected. Note that the node can be put offline if too many protocol errors are reported in succession.
Timeout Errors	Increments by 1 each time the master does not receive a response within the timeout period. The timeout period can be set for each Map Descriptor, each node or for the connection. Default 2 seconds.
Streaming Errors	Increments by 1 each time that the master's serial buffer overflows. This occurs when the master receives messages or bytes faster than it can process. This error will probably only occur with this driver when the messages are malformed for from some other protocol.
Checksum Errors	Increments by 1 each time a complete message is received and the checksum is invalid. Typically indicates a corrupted message.
NAK Errors	The number of times that the master receives a response considered to be a NAK. These responses include device not found, no data, device busy etc. In versions of the driver prior to 1.02a when a NAK error was encountered by the Client it had the effect of slowing down polling.
Node Offline Errors	The number of times the master receives a response reporting the remote node is offline.
IC Timeout Errors	The number of times the Server detects a gap in the byte stream between receiving the first and last bytes of a message. Usually indicates an incomplete message or a dropped byte.
No Start Error	The number of times the Server receives a message which did not start with the correct two byte header. If this stat increments frequently then this probably indicates that the driver is receiving messages from some other protocol.
No Station Errors	The number of times that the driver receives a message for a node that it cannot find or which was offline.
Bad Station Errors	The number of times the Server receives a poll which required a capability not defined in the CSV file. E.g. A message to "Enable Y1" was received but there wasn't a Map Descriptor with its Lennox_Func set to "Enable Y1".
PLC Bytes Received	The number of bytes that the Server has received in complete messages. Bytes from badly formed messages are not counted.
PLC Messages Received	The number of well formed complete messages received by the Server.

Operating Stat	Notes
PLC Read Messages Sent	The number of messages sent by the Server that read data from the remote device.
PLC Write Messages Sent	The number of messages sent by the Server that sent data to the remote device or that requested the remote device perform some command.
PLC Bytes Sent	The number of bytes the Server has sent in read/write messages sent to the remote devices.
Scada Messages Sent	The number of messages sent by the Server.
Scada Bytes Sent	The number of bytes contained in the above messages.
Scada messages Received	The number of complete well formed messages received by the Server.

NAK	Each time a negative acknowledgement message is received.
CHECKSUM	Each time a message is received containing a non-ASCII character.
NOISE	An acknowledgement message containing an unrecognized code.
NO START	Each time a message is received that doesn't begin with a space.
PROTOCOL	All other errors are reported as protocol errors

9.1 Driver Specific Operating Statistics

In addition to the standard FieldServer communication statistics described above and in the FieldServer Instruction Manual, this driver can also expose some driver statistics by writing data to a Data Array. A special Map Descriptor is required. The driver recognizes the Map Descriptor by its name which must be "Lennox-stats". In addition, a special node must be defined. The driver recognizes this node by its name which must be "Lennox-stats-node".

The following example shows how this special Map Descriptor can be configured. You can copy this section of text directly into your CSV file.

```
Nodes
Node_Name,                  , Protocol          , Node_ID
Lennox-stats-node           , Lennox            , 64

Data_Arrays
Data_Array_Name             , Data_Format       , Data_Array_Length
DA_LNNX_STATS               , UINT32            , 1000

Map_Descriptors
Map_Descriptor_Name         , Data_Array_Name   , Node_Name          , Length
Lennox-stats                , DA_LNNX_STATS     , Lennox-stats-node  , 1000
```

When the driver sees this Map Descriptor it uses the Data Array *DA_LNNX_STATS* (in this example) to store driver specific statistics. Only one of these Map Descriptors may be specified per FieldServer.

The driver stores the following data. The location in the Data Array is obtained by multiplying the port number by 100 and then using the location offset indicated in the table below.

Offset	Operating Statistic
1	The number of nodes found with duplicate node numbers
2	The number of times that the Checksum Diagnostic (a QA procedure test) was activated
3	The number of times that the No Message Start Diagnostic (a QA procedure test) was activated
4	The number of times that the No Message End Diagnostic (a QA procedure test) was activated
5	The value (in decimal) of the source address used when the Client last sent a poll
6	The number of times that the Client/Server received a broadcast message
7	The number of times that Server response suppression Diagnostic (a QA procedure test) was activated
8	The number of times that the Server sent a NAK response
9	The type of NAK message (in decimal) of the most recent NAK sent by the Server
10	The number of times the Client expected a response message but received a poll. In cases of 'go offline' this may not be an error.
12	The number of times the Client received a response whose cmd0/1 codes were not recognized. produces a panic.
13	For future use
14	The number of times that the Client or Server received message that had the wrong message header
15	Number of times the Server received a message that it could not parse because the message is a response and not a poll
16	The value of the command code0 of the most recent message the Server received that was a response and not a poll
17	Ditto - except code1
18	Number of times the Server received a message that it could not parse because the message codes are unknown
19	The value of the command code0 of the most recent message the Server received where the message command code was unknown

Offset	Operating Statistic
20	Ditto - except code1
21	Number of times Client received a response to an echo data poll
22	The value of the 1st byte of echoed data from the last echo received
23	Number of times Server received a poll to an echo data poll
24	The value of the 1st byte of echo data from the last echo poll received by the Server
25	The number of times the Server has been polled to change address
26	The value of new address found in the most recent change address poll
27	The number of times the Server has received a poll commanding the unit to go online
28	The number of times the Server has received a poll commanding the unit to go offline
29	The number of times the Server has received a poll requesting a system reset
30	Increments each time the Server receives a poll requesting the error buffer be cleared
31	The number of times the Client receives a response reporting the node is offline
32	The number of times the Client polled the Technician device requesting a device ID
33	The number of times the Technician device did not respond to a Client poll that requested a device ID
34	When a unit claims the bus Server it puts other units offline. This is the address of the unit that most recently put one of the FieldServer nodes offline.
35	The number of times that an offline Server went online because its offline timeout had expired.
36	The number of times that an Client received an offline response
37	The number of times that a Server received poll's while the connection was offline
38	The address of the source of the most recent poll described above
39	The number of times that the Server sends junk bytes before sending the real poll
40	The number of times that the Server found a new Error Buffer Pointer
41	The number of times that the Server Retrieved the 3 error blocks when a new EBP was found
42	The number of times that the Server stored the sorted error table
43	The number of times that the Server found index out of bound(0>index>83) while storing the sorted error table
44	Set to 1 if the most recently polled device did not have its PLC_Type defined
45	Set to 1 if the most recently polled device was a type0 (FUll)
46	Set to 1 if the most recently polled device was a type1 (Reduced 1)
47	Set to 1 if the most recently polled device was a type2 (Reduced 2)
48	Set to 1 if the most recently polled device was a type3 (Minimal)
49	The number of times an 0x06 0x4d response (eeprom busy) as been rcvd
50	The number of times an 0x06 0x4d response was sent
51	Set to 1 if the most recently polled device was a type4 (gateway)
52	Increments each time the driver skips a poll for the error block because optimized err block polling was enabled
53	Increments each time the driver polls for the ebp
54	Increments each time the driver polls for error block 1
55	Stores the source address of the last 'ack' 0x0262 'command executed'
56	Stores the destination address of the last 'ack' 0x0262 'command executed'
57	Increments each time a 'go online' command was received while offline from a station other than the one that put the node offline
58	Increments each time a poll for a Node_ID is sent
59	Increments each time the Server receives a broadcast poll for Node_ID
60	When a new Node_ID is received the driver stores its value here. The previous value is overwritten.
61	When set to 1 then the Server side does not respond at all