



A Sierra Monitor Company

**Driver Manual**  
**(Supplement to the FieldServer Instruction Manual)**

**FS-8700-59 Canatal Satchnet Driver**

**APPLICABILITY & EFFECTIVITY**

**Effective for all systems manufactured after November 2011**

Driver Version: 1.02  
Document Revision: 2

**TABLE OF CONTENTS**

**1 Canatal Satchnet Protocol Driver Description ..... 3**

**2 Driver Scope of Supply ..... 3**

    2.1 Supplied by FieldServer Technologies for this driver.....3

**3 Hardware Connections..... 4**

    3.1 Connection Notes .....5

        3.1.1 *Interconnecting Cable Characteristics* .....5

**4 Data Array Parameters..... 6**

**5 Configuring the FieldServer as a Canatal Satchnet Protocol Driver Client ..... 7**

    5.1 Client Side Connection Descriptor Parameters .....7

    5.2 Client Side Node Descriptor Parameters .....8

    5.3 Client Side Map Descriptor Parameters.....9

        5.3.1 *FieldServer Specific Map Descriptor Parameters* .....9

        5.3.2 *Driver Related Map Descriptor Parameters*.....9

        5.3.3 *Timing Parameters*.....9

        5.3.4 *Map Descriptor Example*.....10

**6 Configuring the FieldServer as a Canatal Satchnet Protocol Driver Server ..... 11**

    6.1 Server Side Connection Parameters .....11

    6.2 Server Side Node Parameters .....12

    6.3 Server Side Map Descriptors.....13

        6.3.1 *FieldServer Specific Map Descriptor Parameters* .....13

        6.3.2 *Driver Specific Map Descriptor Parameters* .....13

        6.3.3 *Timing Parameters*.....13

        6.3.4 *Map Descriptor Example*.....14

**Appendix A. Useful Features..... 15**

    Appendix A.1. Retrieving Data from Tables with Mixed Data Formats .....15

        Appendix A.1.1. *Map Descriptor Example* .....15

**Appendix B. Troubleshooting..... 16**

    Appendix B.1. Improving Communication on Legacy Canatal Devices.....16

    Appendix B.2. Driver Error Messages .....16

**Appendix C. Reference..... 18**

    Appendix C.1. Canatal Device Data Tables .....18

    Appendix C.2. Recommended Data Array Formats Based on Canatal Data Types.....19

## 1 CANATAL SATCHNET PROTOCOL DRIVER DESCRIPTION

The Canatal Satchnet Protocol Driver allows the FieldServer to transfer data to and from devices over either RS-232 or RS-485 using Canatal Satchnet Protocol Driver protocol. The FieldServer can emulate either a Server or Client.

This driver provides support for the Canatal International implementation of the “Satchwell Network (Full) and Net2 (Reduced)” protocol.

The Canatal Satchnet Protocol Driver can act as a Satchnet Master.

- Any Satchnet device on any serial port (P1-P8, R1-R2<sup>1</sup>) can be polled
- Data from Series 2, 4, 5, 6,8, 9 devices can be polled.
- Data from any 'Table' can be read.
- Data in 'Tables' with “write permission” can be written to by the driver.
- The driver can read/write Bit, Byte, Word, BCD and Float Data.

The Canatal Satchnet Protocol Driver can act as a (passive) Server.

- The driver can be polled by a Satchnet Master and return data from the FieldServer's data arrays.
- The driver can emulate Series 2,4,5,6,8,9 devices.
- All “Tables” can be read/written in accordance with their read/write capability

FieldServer Mode	Nodes	Comments
Client	63	Expansion above 32 nodes normally requires a repeater
Server	63	Expansion above 32 nodes normally requires a repeater

## 2 DRIVER SCOPE OF SUPPLY

### 2.1 Supplied by FieldServer Technologies for this driver

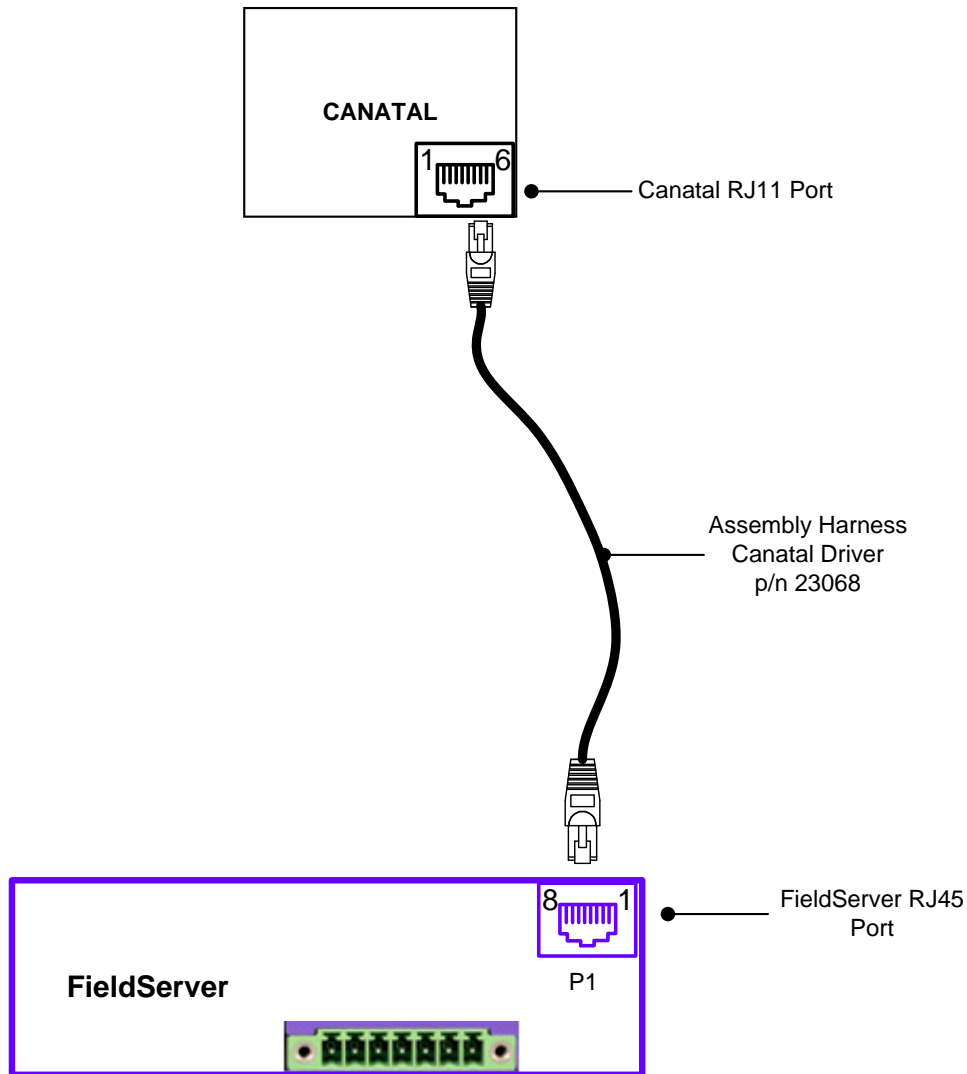
FieldServer Technologies PART #	Description
23068	Assembly Harness – Canatal Driver

<sup>1</sup> Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

### 3 HARDWARE CONNECTIONS

The FieldServer is connected to the Canatal Driver as shown below.

Configure the Canatal Driver according to manufacturer’s instructions.



**Pinouts**

From RJ11	To RJ45	Color
1	6	White
2	1	Black
3	8	Red

## 3.1 Connection Notes

Set Jumper 16, 17 and 18 on the board to position 2-3

### 3.1.1 Interconnecting Cable Characteristics

The interconnecting cable may be composed of twisted or untwisted pair (flat cable) possessing the characteristics below.

- **Conductor Size:** The interconnecting cable shall be composed of two wires of a 24 AWG or larger conductor for solid or stranded copper wires, or for non-copper conductors with sufficient size to yield a DC wire resistance not to exceed 30 ohms per 1000 feet per conductor.
- **Mutual Pair Capacitance:** The capacitance between the two wires in the wire pair shall not exceed 20 picofarads per foot and the value shall be reasonably uniform over the length of the cable.
- **Stray Capacitance:** The capacitance between wires in the cable sheath, with all wires connected to ground shall not exceed 40 picofarads per foot and shall be reasonably uniform over the length of the cable.
- **Pair-to-Pair Balanced Crosstalk:** The balanced crosstalk from one wire pair to any other pair in the same cable sheath shall have a minimum value of 40 decibels of attenuation measured at 150 kilohertz.

An interconnecting cable meeting these specifications will result in a transmission line with nominal characteristic impedance in the order of 100 ohms to frequencies greater than 100 kilohertz and a DC series loop resistance not exceeding 240 ohms.

## 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, UInt16, SInt16, Byte. Refer to Appendix C.2
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.	1-10,000

### 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 CONFIGURING THE FIELDSEVER AS A CANATAL SATCHNET PROTOCOL DRIVER CLIENT**

For a detailed discussion on 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 Canatal Satchnet Client.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for Canatal Satchnet Protocol Driver 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 that in the tables, \* indicates an optional parameter, with the bold legal value being the default.

**5.1 Client Side Connection Descriptor Parameters**

Section Title		
Connections		
Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2 <sup>2</sup>
Baud	Specify baud rate	<b>1200</b> , 2400, 4800, 9600 (Vendor limitation)
Parity	Specify parity	<b>Even</b>
Data_Bits	Specify data bits	<b>7</b>
Stop_Bits*	Specify stop bits	<b>1</b>
Protocol	Specify protocol used	Canatal, Satchnet
Poll Delay*	Time between internal polls	0-32000 seconds, <b>1 s</b>

**Example**

```
// Client Side Connections
Connections
Port , Baud , Parity , Protocol , Data_Bits , Stop_Bits
R1 , 1200 , Even , Canatal , 7 , 1
```

<sup>2</sup> Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

5.2 Client Side Node Descriptor Parameters

Section Title		
Nodes		
Column Title	Function	Legal Values
Node_Name	Provide name for Node	Up to 32 alphanumeric characters
Node_ID	Address of physical server node (Canatal device address)	1-63
Protocol	Specify protocol used	Canatal, Satchnet
Port	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2 <sup>3</sup>
Cana_Node_Protocol_Type*	Specify which specific protocol is supported by the Canatal device. Series 2, 3, 4 and 5 support only "Network". Series 6, 8 and 9 support both "Network" and "Net2". Specifying this as Net2 will significantly improve communication.	Network, Net2

**Example**

```
// Client Side Nodes

Nodes
Node_Name  Node_ID  , Protocol  , Port  , Cana_Node_Protocol_Type
Conditioner1  1      , Canatal  , R1    , Net2
```

<sup>3</sup> 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 Descriptor Parameters

#### 5.3.1 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 Section 4
Data_Array_Location	Starting location in Data Array	0 to (Data_Array_Length-1) as specified in Section 4
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 Section 5.2
Length	Length of Map Descriptor	1 - 1000 (Maximum value is dependent on the values of Cana_Series and Cana_Table) Refer to Appendix C.1
Address	The offset into the Canatal device data table. Each table first element is addressed as zero, the 2nd element as 1 etc.	0,1,2,3 ... The address & length must not exceed the maximum length of the table being addressed.
Cana_Series	Canatal Corporation Inc. Device Series Number.	2,4,5,6,9
Cana_Table	The data table in the device to be read.	Non zero integers. Refer to Appendix C.1
Cana_Format*	For tables with mixed data formats, the driver assigns a default data format. This default can be overridden using this parameter. Refer to Appendix C.1	Byte, BCD, BCD4, Float, Word, Percent, Swap_int.

#### 5.3.3 Timing Parameters

Column Title	Function	Legal Values
Scan_Interval	Rate at which data is polled	>0.1s

### 5.3.4 Map Descriptor Example

In this example two Map Descriptors are provided. The first reads Device 11 (a Series 2 device) and obtains the Alarm Table. The second Map Descriptor reads Device 23 (a series 9 device) and reads the Cooling analog output (2nd element of table 10).

Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Function	Node_Name	Address	Length	Cana_Series	Cana_Table	Scan_Interval
DEV11_ALMS	,ALARMS	,220	,Rdbc	,DEV11	,0	,1	,2	,11	,1.0s
DEV23_COOL	,ANA_OUT	,0	,Rdbc	,DEV23	,1	,1	,9	,10	,1.0s

Use any name that is meaningful. The driver validates Map Descriptors and reports errors referring to the Map Descriptor name.

The first element from table 11 will be placed in element 220 (the 221st element) of the array ALARMS.

The data arrays are defined in the Data\_Arrays part of the CSV. Based on these examples the array called ALARMS should be a byte or integer array as alarms have the value 0,1 or 2 (acked) and it must be at least 239 (220+19) elements in size. The array ANA\_OUT is probably a FLOAT and should have at least 1 element in size.

These Map Descriptors generate read polls on a continuous basis.

The number of items being read. Not the number of bytes being read.

This is the offset into the table being polled. Refer to Canatal International Corporation Inc. Product Literature to determine the required offset.

The polls will occur every second.

You need to know the series of the device you are polling and table number of the table that contains the data of interest. Refer to Appendix C.1

## 6 CONFIGURING THE FIELDSEVER AS A CANATAL SATCHNET PROTOCOL DRIVER SERVER

For a detailed discussion on FieldServer configuration, please refer to the instruction manual for the FieldServer. The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer (See “.csv” files on the driver diskette).

For a detailed discussion on 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 Canatal Satchnet Server.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for (Driver Name) 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 that in the tables, \* indicates an optional parameter, with the bold legal value being the 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-P8, R1-R2 <sup>4</sup>
Baud*	Specify baud rate	<b>1200</b> , 2400, 4800, 9600 (Vendor limitation)
Parity*	Specify parity	<b>Even</b>
Data_Bits*	Specify data bits	<b>7</b>
Stop_Bits*	Specify stop bits	<b>1</b>
Protocol	Specify protocol used	Canatal, Satchnet

**Example**

```

// Server Side Connections

Connections
Port          , Baud          , Parity          , Protocol          , Data_Bits          , Stop_Bits
R1            , 1200           , Even            , Canatal           , 7                  , 1
    
```

<sup>4</sup> Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

## 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	Address of physical server node (Canatal device address)	1-63
Protocol	Specify protocol used	Canatal, Satchnet

### Example

```
// Server Side Nodes

Nodes
Node_Name      , Node_ID  , Protocol
FieldServer   , 11      , Canatal
```

## 6.3 Server Side Map Descriptors

### 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 Section 4.
Data_Array_Location	Starting location in Data Array	0 to (Data_Array_Length-1) as specified in Section 4.
Function	Function of Server Map Descriptor	Passive

### 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 Section 6.2.
Length	Length of Map Descriptor	1 - 1000 (Maximum value is dependent on the values of Cana_Series and Cana_Table) Refer to Appendix C.1
Address	This term is used to refer to the offset into the Canatal device data table. Each table first element is addressed as zero, the 2nd element as 1 etc.	0,1,2,3 ... The address & length must not exceed the maximum length of the table being addressed.
Cana_Series	Refers to the Canatal Corporation Inc. Device Series Number.	2,4,5,6,9
Cana_Table	The data table in the device to be read.	Non zero integers. Refer to Appendix C.1
Cana_Format*	Used to over-ride the default data storage method the driver assigns to the table. This parameter may be required when reading data from a table with mixed data formats. See section 6.1 to identify which tables have mixed data formats and what the default type the driver allocates to the table.	Byte, BCD, BCD4, Float, Word, Percent, Swap_int, -

### 6.3.3 Timing Parameters

Column Title	Function	Legal Values
Scada_Hold_Timeout	Specifies the time the Server side waits before responding to the Client that the Node is offline on the FieldServer Client side.	>1.0s

### 6.3.4 Map Descriptor Example

In this example the FieldServer is emulating a Canatal device addressed as node 11. It is not possible to have the FieldServer emulate two devices of different series addressed as node 11.

Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Function	Node_Name	Address	Length	Cana_Series	Cana_Table
DEV11_TBL5	DA_AI3	0	Passive	Node_12	0	3	4	5

Server side Map Descriptors are passive. They respond to read / write commands from a Canatal master.

The node name connects this Map Descriptor to a node specification.

Corresponds to the offset into the data table being requested.  
 If the request is for offset 2 (3rd element of the table), then the FieldServer responds with data from array DA\_AI3 element 10(offset into array)+3(3rd element).  
 If the request was for offset 5 then this Map Descriptor would not be capable of processing the response. (starts at 0 and is 3 elements long.)

The Canatal device series number the FieldServer will emulate.

The number of the Canatal device's data table the FieldServer will emulate

Appendix A. Useful Features

Appendix A.1. Retrieving Data from Tables with Mixed Data Formats

The driver uses the Data Format listed in Appendix C.1 as the default. If this format is used to store all the data, unexpected results will occur. Read the Canatal product literature, to determine the table storage format of the point(s) of interest and over-ride the default format using the Cana\_Format Parameter.

A separate Map Descriptor is required to poll data of each Data Type where mixed formats are stored in a table.

Appendix A.1.1. Map Descriptor Example.

In this example two Map Descriptors are provided. They both read table 5 of a series 9 device. The default format used by the Driver is Byte, but the data is stored in mixed formats. Thus it is necessary to consult the Canatal literature to determine the format of the required data and specify the Cana\_Format keyword to override the default data format. Refer to Appendix C.1

Reads elements 1 through 7 from table 5 in a series 9 device. According to the product specification the 8th element is stored using a different format and thus using this Map Descriptor to read mixed data type will produce unexpected results.

Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Function	Node_Name	Address	Length	Cana_Series	Cana_Table	Scan_Interval	Cana_Format
TABLE05_1	, TBL05_DATA	, 0	, Rdbc	, DEV11	, 1	, 7	, 9	, 5	, 1.0s	, -
TABLE05_2	, TBL05_DATA	, 0	, Rdbc	, DEV11	, 10	, 1	, 9	, 5	, 1.0s	, Swap_Int

With the offset equal to zero the data is placed in the array element equal to the address. The offset is added to the address when determining the element of the data array to be used for storage.

This is the offset of the Boiler Limit Time in Table 5.

This parameter over-rides the default data type. According to the product specification the **Boiler Limit Time** is stored as a **Swapped Integer**

Appendix B. Troubleshooting

Appendix B.1. Improving Communication on Legacy Canatal Devices

For Canatal Device Series 2-5, setting the Line\_Drive\_Off connection parameter to 0 helps to reduce the number of errors on connection and could improve communication.

Appendix B.2. Driver Error Messages

The following messages are produced by the driver. Those marked *FYI* are for the user's information only and do not require any corrective action. Those marked *Error* require some action from the user.

Error #	Message	Description
CANA:#1	FYI. Station illegal -Forcing from %d to 59	Valid station numbers are 1-63.
CANA:#2	Error. Series & table numbers are reqd. Correct MapDesc=<%s>	The Map Descriptor parameters Cana_Series and Cana_Table need to be specified.
CANA:#3	Error. Illegal Series Number. Correct MapDesc=<%s>	The series number specified cannot be processed by this driver. Read the vendor documentation and correct the CSV file.
CANA:#4	Error. Illegal Table Number(%d). Correct MapDesc=<%s>	All Canatal equipment series do not have the same number of tables. Read the vendor documentation and correct the CSV file.
CANA: #5	Error. Series & table numbers are reqd. Empty Table. Correct MapDesc=<%s>	The series & table combinations specified are not valid. Read the vendor documentation and correct the CSV file.
CANA:#6	Error. Start address is beyond end of table. MapDesc=<%s>	The address is the equivalent term to offset in the Vendor literature. Different tables have different lengths. The lengths may also be different based on the series number of the equipment. Read the vendor documentation and correct the CSV file.
CANA:#7	FYI. Requested too many elements from table. MapDesc=<%s> Changed request from %u to %u	The driver has truncated the Map Descriptor length so that it does not read/write beyond the end of a table
Errors 8 - 12 are produced when parsing a response from a Canatal device. They may result from a corrupted message. If they occur infrequently then they require no attention. Check the communication stats.		
CANA:#8	Error. Poll(%d)/Response(%d) Stns Not Equal	The station polled has responded but as a different station. The response will be abandoned. This may mean that the device's address has been incorrectly set or that the device is not functioning correctly.
CANA:#9	Error. Poll(%d)/Response(%d) Tables Not Equal	The table polled has responded but as a different table. The response will be abandoned. This may mean that the device's address has been incorrectly set or that the device is not functioning correctly.
CANA:#10	Error. Response Table=%d Unknown	The table number specified cannot be processed by this driver. Read the vendor documentation and correct the CSV file.

Error #	Message	Description
CANA:#11	Error. Poll(%d)/Response(%d) Address Not Equal	The address polled has responded but as a different address. The response will be abandoned. This may mean that the device's address has been incorrectly set or that the device is not functioning correctly.
CANA:#12	Error. Response Expected(%d)/Actual(%d) Bytes	The data was requested from a table with mixed data formats and the data served was not in Byte (default) format. Use the Cana_Format parameter to override the Data Format in the CSV file.
CANA:#13	Error. Array is too small for mapDesc=<%s> Some data will not be stored.	This is regarded as an error as the Map Descriptor has requested more data than can be stored. The excess data has been discarded. Modify the Map Descriptor definition or change the array length in the CSV file and restart the FieldServer.
CANA:#14	Error. Unknown Data method (%d) for mapDesc=<%s>	Report this error to FIELDSEVER TECHNOLOGIES.
CANA:#15	Error. Unknown Data method (%d) for mapDesc=<%s>	
CANA:#16	Error. Ack Expected	

Appendix C. Reference

Appendix C.1. Canatal Device Data Tables

Canatal Devices have their data organized into data tables. A query requests a number of elements from a particular data table. The number, length and content of data tables vary according to the series number of the device being addressed. The following series and tables are supported by this driver.

Note that for Canatal firmware revisions from 2002 onwards, "Time" data available in Table 13 is now available in Table 69, and Table 13 has been made obsolete.

Table	Data Format	Max Elements	Read/Write	Description	Notes
<b>Series 2</b>					
4	CANA_BCD_CODED	6	R	FirmWare	
5	CANA_FLOAT	34	RW	Keycode	
6	CANA_FLOAT	6	R	Sensor Reading	
7	CANA_FLOAT	3	RW	Security	
8	CANA_BYTE	84	RW	Time Schedule	
9	CANA_BYTE	2	R	Switched Output	
10	CANA_PERCENT	6	R	Analog Outputs	
11	CANA_BYTE	19	RW	Alarm Status	
12	CANA_BYTE	1	R	Control Status	
13	CANA_BCD_CODED	7	RW	Time	
14	CANA_BYTE	4	R	Staged Output Status	
15	CANA_BYTE	1	RW	Plant Status	
<b>Series 4</b>					
4	CANA_BCD_CODED	6	R	FirmWare	
5	CANA_FLOAT	34	RW	Keycode	
6	CANA_FLOAT	6	R	Sensor Reading	
7	CANA_FLOAT	3	RW	Security	
8	CANA_BYTE	1	R	Switched Output	
9	CANA_PERCENT	5	R	Analog Outputs	
10	CANA_BYTE	16	RW	Alarm Status	
11	CANA_BYTE	1	R	Control Status	
12	CANA_BYTE	4	R	Staged Output Status	
13	CANA_BYTE	1	RW	Plant Status	
<b>Series 5</b>					
4	CANA_BCD_CODED	6	R	FirmWare	
5	CANA_SWAP_INT	100	RW	Keycode	
6	CANA_BYTE	4	R	Sensor Reading	
7	CANA_SWAP_INT	3	RW	Security	
8	CANA_BYTE	84	RW	Time Schedule	
9	CANA_BYTE	3	R	Switched Output	
10	CANA_PERCENT	4	R	Analog Outputs	
11	CANA_BYTE	48	RW	Alarm Status	
12	CANA_BYTE	3	R	Control Status	
13	CANA_BCD_CODED	7	RW	Time	
14	CANA_BYTE	4	R	Staged Output Status	
15	CANA_WORD	17	RW	Accumulating Run Time	
15	CANA_BYTE	1	RW	Machine Remote Control	
<b>Series 6 &amp; Series 8</b>					

Table	Data Format	Max Elements	Read/Write	Description	Notes
4	CANA_BCD_CODED	6	R	FirmWare	
5	CANA_BYTE	75	RW	Keycode	Mixed
6	CANA_SWAP_INT	7	R	Sensor Reading	
7	CANA_BCD_4	3	RW	Security	
8	CANA_BYTE	84	RW	Time Schedule	
9	CANA_BYTE	16	R	Switched Output	
10	CANA_PERCENT	4	R	Analog Outputs	
11	CANA_BYTE	241	RW	Alarm Status	
12	CANA_BYTE	10	R	Control Status	
13	CANA_BCD_CODED	7	RW	Time	
14	CANA_BYTE	4	R	Staged Output Status	
15	CANA_SWAP_INT	64	RW	Accumulating Run Time	
16	CANA_BYTE	1	RW	Machine Remote Control	
17	CANA_BYTE	7	RW	KeyCode2	Mixed
<b>Series 9</b>					
4	CANA_BCD_CODED	6	R	FirmWare	
5	CANA_BYTE	75	RW	Keycode	Mixed
6	CANA_SWAP_INT	7	R	Sensor Reading	
7	CANA_BCD_4	3	RW	Security	
8	CANA_BYTE	84	RW	Time Schedule	
9	CANA_BYTE	16	R	Switched Output	
10	CANA_PERCENT	4	R	Analog Outputs	
11	CANA_BYTE	121	RW	Alarm Status	
12	CANA_BYTE	10	R	Control Status	
13	CANA_BCD_CODED	7	RW	Time	
14	CANA_BYTE	4	R	Staged Output Status	
15	CANA_SWAP_INT	64	RW	Accumulating Run Time	
16	CANA_BYTE	1	RW	Machine Remote Control	
17	CANA_BYTE	7	RW	KeyCode2	Mixed

Appendix C.2. Recommended Data Array Formats Based on Canatal Data Types

Data Type	Suitable Data Array Format
CANA_BCD_CODED	Byte, Uint16, Uint32, Float
CANA_BYTE	Byte, Uint16, Uint32, Float
CANA_SWAP_INT	Sint16, Sint32, Uint16, Uint32, Float
CANA_BCD_4	Byte
CANA_WORD	Uint16, Uint32, Float
CANA_FLOAT	Float