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 May 1, 2001

Instruction Manual Part Number FS-8700-59

3/13/03

TABLE OF CONTENTS

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 Corporation implementation of the "Satchwell Network" protocol.

The Canatal Satchnet Protocol Driver can act as a Master.

- Any Satchnet device on any serial port (P1-P8, R1 & R2 on the FS-B40 Series, serial port on the FS-B20 series) can be polled.
- Data from Series 2, 4, 5, 6, 9 devices can be polled.
- Data from any 'Table' can be read.
- Data in 'Tables' with write permission can be written by the driver.
- The driver can read/write Bit, Byte, Swapped Integer, 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 2,4,5,6,9 devices.
- All 'Tables' can be read / written.

2. Hardware/Software

2.1 Supplied by FieldServer Technologies for this driver

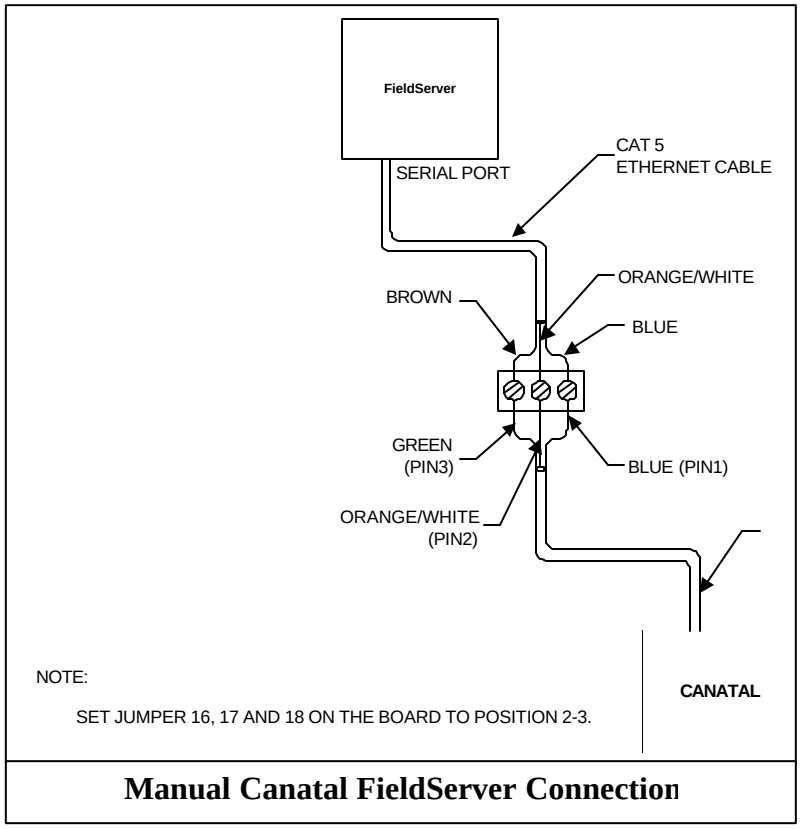
RS-485 connection adapter
Driver Manual.

2.2 Provided by user

Canatal Device
UTP or similar cable with RJ11 adapter.

3. Hardware Connections

The FieldServer is connected to the as shown below.
Configure the Canatal Driver according to manufacturer's instructions.



Manual Canatal FieldServer Connection

Canatal Corporation Inc. have specified the following data with respect to interconnecting cables.

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 one wire in the pair to the other wire 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 one wire in the cable to all others in the cable sheath, with all others 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 pair of wire 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 a 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. Configuring the FieldServer as a Canatal Satchnet Protocol Driver Client

For a detailed discussion on FieldServer configuration, please refer to the Configuration 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).

This section documents and describes the parameters necessary for configuring the FieldServer as a master, configured to communicate with a Canatal Satchnet Protocol Driver Server (a passive Canatal device).

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.

4.1 Data Arrays

Section Title
Data_Arrays

Column Title	Function	Legal Values
Data_Array_Name	Provide name for Data Array	Up to 15 alphanumeric characters
Data_Format	Provide data format. Each data array can only take on one format.	FLOAT, BIT, UInt16, SInt16, Packed_Bit, Byte, Packed_Byte, Swapped_Byte
Data_Array_Length	Number of Data Objects. Must be larger than the data storage area required for the data being placed in this array.	1-10,000

Example

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

4.2 Client Side Connections

Section Title		
Connections		
Column Title	**Function**	**Legal Values**
Port	Specify which port the device is connected to the FieldServer	FS x40: P1-P8, R1-R2 FS x20: P1-R2
Baud	Specify baud rate	1200
Parity	Specify parity	Even
Data_Bits	Specify data bits	7
Stop_Bits*	Specify stop bits	**1**
Protocol	Specify protocol used	Canatal, Satchnet
Handshaking*	Specify hardware handshaking	RTS, RTS/CTS, **None**
Poll Delay*	Time between internal polls	0-32000 seconds **default 1 second**

Example

```
//    Client Side Connections

Connections
Port, Baud, Parity, Protocol, Data_bits, Stop_bits,
R1,   1200, Even,   Canatal,  7,         1,
```

4.3 Client Side Nodes

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	FS x40: P1-P8, R1-R2 FS x20: P1-R2

Example

```
//    Client Side Nodes

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

4.4 Client Side Map Descriptors

4.4.1 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_Location	Starting location in Data Array	0 to maximum specified in "Data Array" section above
Function	Function of Client Map Descriptor	RDBC, WRBC, WRBX

4.4.2 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 "Client Node Descriptor" above
Length	Length of Map Descriptor	1 - 1000 (Maximum value is dependent on the values of Cana_Series and Cana_table. For more information see section 6.1
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. The address & length must not exceed the maximum length of he table being addressed.	0,1,2,3 ...
Protocol Specific Parameters		
Cana_Series	Refers to the Canatal Corporation Inc. device series number. Although different series of equipment have similar tables with similar contents they are not all the same, the contents may vary and the data formats do vary.	2,4,5,6,9
Cana_Table	This is the data table in the device you wish to read.	Non zero integers. Legal values may be obtained by reading section 6.1.
Cana_Format *	Used to over-ride the default data storage method the driver assigns to the table. This paramater may be required when reading data from a table with midex data formats. See section 6.1 to identify which table have mixed data formats and what the default type the driver allocates to the table.	Bype, BCD, BCD4, Float, Word, Percent, Swap_int.

4.4.3 Timing Parameters

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

4.4.4 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     ,19     , 2          ,11          , 1.0s
DEV23_COOL      ,ANA_OUT     ,0              , rdbc   ,DEV23   , 1     ,1      , 9          ,10          , 1.0s
```

Any name that is meaningful to you. 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 node names point the driver at a Node definition which specifies a port and a device address.

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

The polls will ocur every second.

The data arrays are defined in the Data_Arrays part of the CSV. Based on these exampled 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.

This is the offset into the table being polled.

Determine the offset you should use by reading the Canatal International Corporation Inc. Product Literature.

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

Read section 6.1 for more information or read the vendor product literature.

4.4.5 Map Descriptor Example 2

In this example two map descriptors are provided. They both read table 16 of a series 9 device. You will see though that the 2nd map descriptor has an extra keyword to override the default format. Read section 6.1 form more information on default data formats.

// 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_Descriptors
Map_Descriptor_Name, Data_Array_Name, Data_Array_Offset, Function, node_name, Address, Length, Cana_Series, Cana_table, Scan_Interval
TABLE05_1           ,TBL05_DATA   ,0                    ,  rdbc    , DEV11   , 1       , 6       , 9             , 5           , 1.0s
```

*// Reads the **Boiler Limit Time** (Offset 10) from table 5 in a series 9 device According to the product specification the **Boiler Limit Time** is stored as a **Swapped Integer**. The Default format this driver uses for the table is **Byte**.*

```
Map_Descriptors
Map_Descriptor_Name, Data_Array_Name, Data_Array_Offset, Function, node_name, Address, Length, Cana_Series, Cana_table, Cana_format, Scan_Interval
TABLE05_2           ,TBL05_DATA   , 0                   ,  rdbc    , DEV11   , 7       , 1       , 9             , 5           , swap_int    , 1.0s
```

Any name that is meaningful to you. The driver validates map descriptors and reports errors referring to the map descriptor name.

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. It is only required if you need to over-ride the driver's default.

5. Configuring the FieldServer 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).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a Canatal Satchnet Protocol Driver Client (Master)

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 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 that in the tables, * indicates an optional parameter, with the bold legal value being the default.

5.1 Data Arrays

Section Title		
Data_Arrays		
Column Title	**Function**	**Legal Values**
Data_Array_Name	Provide name for Data Array	Up to 15 alphanumeric characters
Data_Format	Provide data format. Each data array can only take on one format.	FLOAT, BIT, UInt16, SInt16, Packed_Bit, Byte, Packed_Byte, Swapped_Byte
Data_Array_Length	Number of Data Objects. Must be larger than the data storage area required for the data being placed in this array.	1-10,000

Example

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

5.2 Server Side Connections

Section Title		
Connections		
Column Title	**Function**	**Legal Values**
Port	Specify which port the device is connected to the FieldServer	FS x40: P1-P8, R1-R2 FS x20: P1-R2
Baud*	Specify baud rate	110 – 115200, standard baud rates only
Parity*	Specify parity	Even, Odd, **None,** Mark, Space
Data_Bits*	Specify data bits	7, **8**
Stop_Bits*	Specify stop bits	**1**
Protocol	Specify protocol used	Canatal
Handshaking*	Specify hardware handshaking	RTS, RTS/CTS, **None**

The FieldServer is capable of handling any of the above connection specification but the Canatal Corporation Inc. devices are support the specification provided in the example below.

Example

```
//    Server Side Connections

Connections
Port, Baud, Parity, Protocol, Data_bits, Stop_bits,
R1,   1200, Even,   Canatal,  7,         1,
```

5.3 Server Side Nodes

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
```

5.4 Server Side Map Descriptors

5.4.1 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_Location	Starting location in Data Array	0 to maximum specified in "Data Array" section above
Function	Function of Server Map Descriptor	PASSIVE

5.4.2 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 "Client Node Descriptor" above
Length	Length of Map Descriptor	1 - 1000 (Maximum value is dependent on the values of Cana_Series and Cana_table. For more information see section 6.1
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. The address & length must not exceed the maximum length of he table being addressed.	0,1,2,3 ...
Protocol Specific Parameters		
Cana_Series	Refers to the Canatal Corporation Inc. device series number. Although different series of equipment have similar tables with similar contents they are not all the same, the contents may vary and the data formats do vary.	2,4,5,6,9
Cana_Table	This is the data table in the device you wish to read.	Non zero integers. Legal values may be obtained by reading section 6.1.
Length	Length of Map Descriptor	1 - 1000 (Maximum value is dependent on the values of Cana_Series and Cana_table. For more information see section 6.1
Cana_Format *	Used to over-ride the default data storage method the driver assigns to the table. This paramater may be required when reading data from a table with midex data formats. See section 6.1 to identify which table have mixed data formats and what the default type the driver allocates to the table.	Bype, BCD, BCD4, Float, Word, Percent, Swap_int.

5.4.3 Timing Parameters

Column Title	Function	Legal Values
Scada_Hold_Timeout	Specifies time server side waits before responding to client that node is offline on FieldServer client side.	>1.0s

5.4.4 Map Descriptor Example

In this example the FieldServer answers queries from a Canatal Satchnet master device. A same map descriptor could be used when the same master wants to write data to the FieldServer. 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 difefrent 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 cabaple of processing the response. (starts at 0 and is 3 elements long.)

Read section 6.1 for more information.

The FieldServer needs to know which Canatal device series number to emulate.

This is the number of the Canatal device's data table you wish the FieldServer to emulate.

6. Driver Notes

6.1 Canatal Device Data Tables

Canatal Devices's have their data organized into data tables. A query requests a number of elements from a particular data table. The number, length and contents of the data tables is defined by the Canatal Corporation and is dependent on the series number of the device being addressed.

You will need to be familiar with the data tables you with to poll usin g this driver.

The following table specifies the series and tables that can be polled / written to / emulated to this driver.

Series	Table	Data Format	Max Elements	Read Write	Description	Notes
2	4	CANA_BCD_CODED	6	R	FirmWare	
2	5	CANA_FLOAT	34	RW	Keycode	
2	6	CANA_FLOAT	6	R	Sensor Reading	
2	7	CANA_FLOAT	3	RW	Security	
2	8	CANA_BYTE	84	RW	Time Schedule	
2	9	CANA_BYTE	2	R	Switched Output	
2	10	CANA_PERCENT	6	R	Analog Outputs	
2	11	CANA_BYTE	19	RW	Alarm Status	
2	12	CANA_BYTE	1	R	Control Status	
2	13	CANA_BCD_CODED	7	RW	Time	
2	14	CANA_BYTE	4	R	Staged Output Status	
2	15	CANA_BYTE	1	RW	Plant Status	
4	4	CANA_BCD_CODED	6	R	FirmWare	
4	5	CANA_FLOAT	34	RW	Keycode	
4	6	CANA_FLOAT	6	R	Sensor Reading	
4	7	CANA_FLOAT	3	RW	Security	
4	8	CANA_BYTE	1	R	Switched Output	
4	9	CANA_PERCENT	5	R	Analog Outputs	
4	10	CANA_BYTE	16	RW	Alarm Status	
4	11	CANA_BYTE	1	R	Control Status	
4	12	CANA_BYTE	4	R	Staged Output Status	
4	13	CANA_BYTE	1	RW	Plant Status	
5	4	CANA_BCD_CODED	6	R	FirmWare	
5	5	CANA_SWAP_INT	100	RW	Keycode	
5	6	CANA_BYTE	4	R	Sensor Reading	
5	7	CANA_SWAP_INT	3	RW	Security	
5	8	CANA_BYTE	84	RW	Time Schedule	
5	9	CANA_BYTE	3	R	Switched Output	

Series	Table	Data Format	Max	Read	Description	Notes
5	10	CANA_PERCENT	4	R	Analog Outputs	
5	11	CANA_BYTE	48	RW	Alarm Status	
5	12	CANA_BYTE	3	R	Control Status	
5	13	CANA_BCD_CODED	7	RW	Time	
5	14	CANA_BYTE	4	R	Staged Output Status	
5	15	CANA_WORD	17	RW	Accumulating Run Time	
5	15	CANA_BYTE	1	RW	Machine Remote Control	
6	4	CANA_BCD_CODED	6	R	FirmWare	
6	5	CANA_BYTE	75	RW	Keycode	Mixed
6	6	CANA_SWAP_INT	7	R	Sensor Reading	
6	7	CANA_BCD_4	3	RW	Security	
6	8	CANA_BYTE	84	RW	Time Schedule	
6	9	CANA_BYTE	16	R	Switched Output	
6	10	CANA_PERCENT	4	R	Analog Outputs	
6	11	CANA_BYTE	241	RW	Alarm Status	
6	12	CANA_BYTE	10	R	Control Status	
6	13	CANA_BCD_CODED	7	RW	Time	
6	14	CANA_BYTE	4	R	Staged Output Status	
6	15	CANA_SWAP_INT	64	RW	Accumulating Run Time	
6	16	CANA_BYTE	1	RW	Machine Remote Control	
6	17	CANA_BYTE	7	RW	KeyCode2	Mixed
9	4	CANA_BCD_CODED	6	R	FirmWare	
9	5	CANA_BYTE	75	RW	Keycode	Mixed
9	6	CANA_SWAP_INT	7	R	Sensor Reading	
9	7	CANA_BCD_4	3	RW	Security	
9	8	CANA_BYTE	84	RW	Time Schedule	
9	9	CANA_BYTE	16	R	Switched Output	
9	10	CANA_PERCENT	4	R	Analog Outputs	
9	11	CANA_BYTE	121	RW	Alarm Status	
9	12	CANA_BYTE	10	R	Control Status	
9	13	CANA_BCD_CODED	7	RW	Time	
9	14	CANA_BYTE	4	R	Staged Output Status	
9	15	CANA_SWAP_INT	64	RW	Accumulating Run Time	
9	16	CANA_BYTE	1	RW	Machine Remote Control	
9	17	CANA_BYTE	7	RW	KeyCode2	Mixed

Mixed Data Formats: The drivers uses the Data Format listed in the above table as the default. If you wish to poll data of different data types from one table then you will need at least two map descriptors for that table. Example 2 in section 4.4.4 illustrates this. You will use the *cana_format* parameter in the map descriptor to over-ride the default. Read the Canatal product literature, determine the table storage format of the point(s) of interest and over-ride the default format in your map descriptor. Note that map descriptors that include a range of points of mixed data formats will cause unexpected results.

The Data formats are used internally in the driver but they do indicate to the user the type of data array that should be used for storage in the FieldServer.

Data Type	Suitable Data Array Types
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

6.2 Troubleshooting

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.

CANA:#1 FYI. Station illegal -Forcing from %d to 59
Valid station numbers are 1-63.

CANA:#2 Error. Series & table numbers are rqd. Correct MapDesc=<%s>
The series and table parameters cana_series and cana_table must be specified in the map descriptors.

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 rqd. Empty Table. Correct MapDesc=<%s>
The series & table combination 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 you data_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. In fact, it is more likely that you will be alerted to them by seeing errors in the communication statistics.

CANA:#8 Error. Poll(%d)/Response(%d) Stns Not Equal
The station polled has response 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

CANA:#10 Error. Response Table=%d Unknown

CANA:#11 Error. Poll(%d)/Response(%d) Address Not Equal

CANA:#12 Error. Response Expected(%d)/Actual(%d) Bytes

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. The user should 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 FIELDSERVER TECHNOLOGIES.

CANA:#15 Error. Unknown Data method (%d) for mapDesc=<%s>
Report this error to FIELDSERVER TECHNOLOGIES.

7. Version Control

Date	Driver Version	Document Revision	Responsible	Comment
	1.00a	0		Initial Release
	1.00b	1		Minor changes in the way *offset* was described in the examples in section 4.4.4 and 5.4.4
				Driver changes are described in Technical Bulletin TB0001.
5Apr2002	1.00b	2		Corrected some table numbers in section 6.1. Previously table 16 & 17 had been shown as table 15 & 16.
11/22/02	1.00b	3	MF	Formatting changes
3/13/03	1.00b	4	JD	Formatting changes