



A Sierra Monitor Company

Driver Manual
(Supplement to the FieldServer Instruction Manual)
FieldServer-8700-71 TAC/CSI I/Net®2000

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after May 1, 2001

Driver Version:	1.05
Document Revision:	2

TABLE OF CONTENTS

1.	TAC-I/NET DESCRIPTION	3
2.	DRIVER SCOPE OF SUPPLY	4
2.1.	Supplied by FieldServer Technologies for this driver	4
2.2.	Provided by the Supplier of 3 rd Party Equipment.....	4
2.2.1.	<i>Hardware</i>	4
3.	HARDWARE CONNECTIONS.....	5
3.1.	Hardware Connection Tips / Hints.....	5
4.	CONFIGURING THE FIELDSEVER AS A TAC-I/NET CLIENT	6
4.1.	Data Arrays/Descriptors.....	6
4.2.	Client Side Connection Descriptions	7
4.3.	Client Side Node Descriptors.....	7
4.4.	Client Side Map Descriptors.....	8
4.4.1.	<i>FieldServer Related Map Descriptor Parameters</i>	8
4.4.2.	<i>Driver Related Map Descriptor Parameters</i>	8
4.4.3.	<i>Timing Parameters</i>	8
4.4.4.	<i>Map Descriptor Example.</i>	9
5.	CONFIGURING THE FIELDSEVER AS A TAC-I/NET SERVER.....	10
5.1.	Server Side Connection Descriptors	10
5.2.	Server Side Node Descriptors.....	11
5.3.	Server Side Map Descriptors	12
5.3.1.	<i>FieldServer Specific Map Descriptor Parameters</i>	12
5.3.2.	<i>Driver Specific Map Descriptor Parameters</i>	12
5.3.3.	<i>Map Descriptor Example.</i>	13
APPENDIX A.	DRIVER NOTES	14
Appendix A.1.	Scaling.....	14
Appendix A.2.	I/Net Command Table.....	14
Appendix A.3.	Interpreting Server Driver Error Statistics.....	15

1. TAC-I/Net Description

The TAC-I/Net driver allows the FieldServer to transfer data to and from Server and Client devices on a MR LAN over RS-485 using TAC-I/Net driver protocol.

The protocol works over a multidrop RS-485 line and allows only one Client and up to 32 Servers on a network channel. Servers are assigned station addresses that range from 0 to 31. The Client does not have a station address.

If more than 320 points are needed on the I/net network an alternative is for the customer to purchase the 7797-15 Modbus RTU interface card. This interface card will allow 2000 points to be passed to the I/net Sublan. This will change the application from I/net to Modbus, but it will work.

Max Nodes Supported

FieldServer Mode	Nodes	Comments
Client	1	Only 1 client node allowed on multidrop systems
Server	32	This is the limit per i/net panel. The 32 nodes correspond to the maximum of 32 mr's that an i/net panel supports.

The protocol is strictly command/response with the Client polling the Server. Servers can only be MR devices. MR devices contain points that can be polled by the Client.

The driver supports the configuration of the following I/Net type points:

Analog inputs	Analog outputs	Digital inputs	Digital outputs
Analog Input (AI)	Analog Output (AO)	Discrete Indicate (DI)	Discrete Output (DO)
		Discrete Alarm (DA)	Discrete Control (DC)
		Discrete Monitor (DM)	

Both Client and Server drivers support quick polling as well as write ability.

2. Driver Scope of Supply

2.1. Supplied by FieldServer Technologies for this driver

FieldServer Technologies PART #	Description
FS-8917-16	10BaseT Crossover Cable, 7 foot.
FS-8700-71	Driver Manual.

2.2. Provided by the Supplier of 3rd Party Equipment

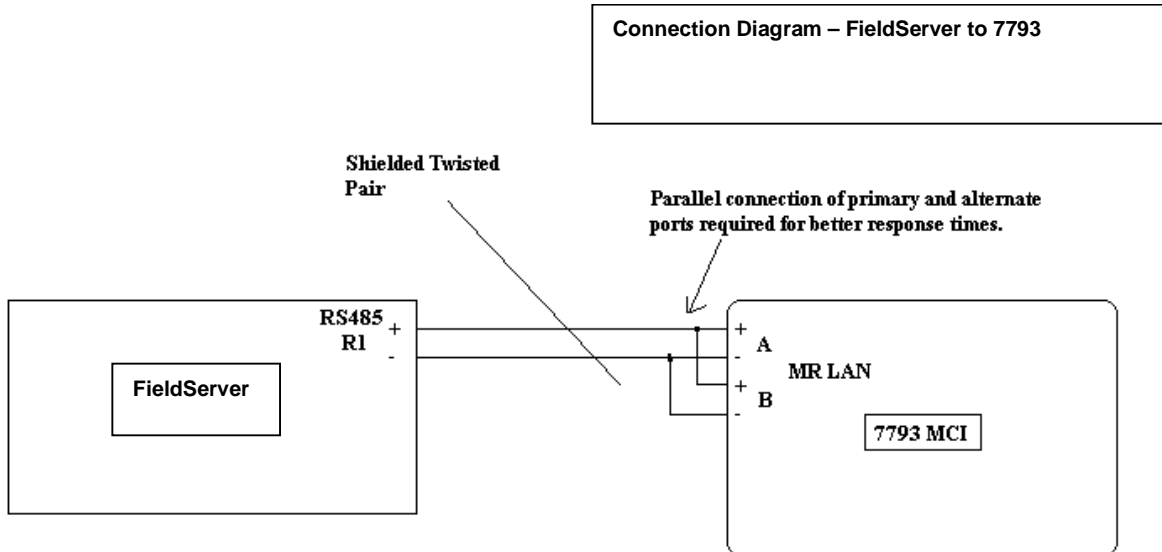
2.2.1. Hardware

Part #	Description
	TAC 7793 (or equivalent) controller

3. Hardware Connections

Configure the devices according to manufacturer's instructions.

Connection to an I/Net device can be depicted as follows:



3.1. Hardware Connection Tips / Hints

The TAC-I/Net Client driver supplied with the FieldServer is currently only used for emulating TAC-I/Net hardware such as the 7793 MCI and should not be used under normal circumstances. Please refer to the section on how to configure the FieldServer as a TAC-I/Net Server if you intend to connect the FieldServer to TAC-I/Net hardware as shown in the connection diagram above. TAC-I/Net hardware such as the 7793 MCI are Client devices to the FieldServer since they initiate polls with commands to the FieldServer which acts as a Server.

It is recommended that the FieldServer have a dedicated i/net port and a dedicated 7793 controller to avoid compatibility problems with other devices on the network.

4. Configuring the FieldServer as a TAC-I/Net 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 TAC-I/Net Server

4.1. Data Arrays/Descriptors

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for TAC-I/Net 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.

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.	Uint16, bit, packed_bit
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_Format,	Data_Array_Length
An_inputs_0,	Uint16,	10
An_outputs_0,	Bit,	10

4.2. Client Side Connection Descriptions

Section Title		
Connections		
Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer	R1-R2 ¹
Protocol	Specify protocol used	I/Net
Baud*	Specify baud rate	9600 Vendor limitation – FieldServer supports all standard baud rates.
Parity*	Specify parity	None,
Data_Bits*	Specify data bits	8
Stop_Bits*	Specify stop bits	1
Handshaking*	Specify hardware handshaking	None

Example

```
// Client Side Connections

Connections
Port, Protocol, Baud, Parity, Handshaking
R1, I/Net, 9600, None, None
```

4.3. 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	Station number of MR to poll	0-31
Protocol	Specify protocol used	I/Net
Port	Specify which port the device is connected to the FieldServer	R1-R2 ¹
Node_Type*	Can be used to disable the periodic broadcast of date and time messages not supported by some devices such as the MR88	date_time_disabled

Example

```
// Client Side Nodes

Nodes
Node_Name, Node_ID, Protocol, Port
MR_00, 0, I/Net, R1
```

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

4.4. Client Side Map Descriptors

4.4.1. 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, PASSIVE

4.4.2. 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 "Client Node Descriptor" above
Address	The I/Net start point number	0 – 9
Length	The number of I/Net points this map descriptor will scan starting from the start point number	1 – 10
Inet_point_type	The type of I/Net point(s) this map descriptor points to.	Analog_input Analog_output Discrete_indicate Discrete_alarm Discrete_monitor Discrete_output Discrete_control
Quick_poll	Specifies whether the points on this node should be quick polled or not.	yes, no (omitting this field will default to quick polling disabled for this node)

4.4.3. Timing Parameters

Column Title	Function	Legal Values
Scan_Interval	Rate at which data is polled	≥1.0s

4.4.4. Map Descriptor Example.

```
// Client Side Map Descriptors
```

Map_Descriptor_Name,	Scan_Interval,	Data_Array_Name,	Data_Array_Offset,	Function,	Node_name,	Inet_point_type,	Address,	Length
MR00_inputs_0,	1s,	An_inputs_0,	0,	RDBC,	MR_00,	analog_input,	0,	1
MR00_inputs_1,	0s,	An_inputs_0,	1,	Passive,	MR_00,	analog_input,	1,	1

This can be any name but each name must be unique. Name will appear in FieldServer Map Descriptor status information screens.

The rate at which the M will be polled

The Data_Array_Name must be one found under Data_Arrays. Data from the scan will be stored into the array at Data_Array_Offset

This value specifies the offset into the data array where the data fetched will be stored.

Write, Read or passive allowed. Only allow one read function per node Map Descriptor to prevent double or triple polling a node.

Node name must be one found under Nodes, Node_Name. The Node_Name identifies the MR station address to be polled.

The type of I/NET data this Map Descriptor points to

The starting point number.

The number of points

5. Configuring the FieldServer as a TAC-I/Net Server

For a detailed discussion on FieldServer configuration, please refer to the FieldServer instruction manual (Virtual PLC). 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 TAC-I/Net Client.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for TAC-I/Net 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. Server Side Connection Descriptors

Section Title		
Connections		
Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer	R1-R2 ²
Protocol	Specify protocol used	I/Net
Baud*	Specify baud rate	9600
Parity*	Specify parity	None
Data_Bits*	Specify data bits	8
Stop_Bits*	Specify stop bits	1
Handshaking*	Specify hardware handshaking	None
Turnaround_delay	Time to wait before responding	0s
Line_drive_on	Time to wait before driving line	0.001s
Line_drive_off	Time after transmit to float line	0s

Example

//	Server Side Connections							
Connections								
Port,	Protocol,	Baud,	Parity,	Handshaking,	Turnaround_delay,	Line_drive_on,	Line_drive_off	
R1,	I/Net,	9600,	None,	None,	0s,	0.001s,	0s	

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.

5.2. Server Side Node Descriptors

Section Title		
Nodes		
Column Title	Function	Legal Values
Node_Name	Provide name for node	Up to 32 alphanumeric characters
Node_ID	The MR station number	0-31
Protocol	Specify protocol used	I/Net
Port	The hardware port this node is on	R1-R2 ³

Example

// Server Side Nodes			
Nodes			
Node_Name,	Node_ID,	Protocol,	Port
MR_00,	0,	I/Net,	R1

³ 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. Server Side Map Descriptors

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 "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	Passive
Data_Array_Low_Scale*	Lower end of scale	0 – 65535
Data_Array_High_Scale*	Higher end of scale	0 – 65535
Node_Low_Scale*	Lower end of scale	0 – 65535
Node_High_Scale*	Higher end of scale	0 – 65535

5.3.2. 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 "Client Node Descriptor" above
Inet_point_type	The I/Net point type this map descriptor points to.	Analog_input Analog_output Discrete_indicate Discrete_alarm Discrete_monitor Discrete_output Discrete_control
Length	The number of I/Net points starting from the start point number this map descriptor points to.	1 – 10
Address	The I/Net start point number.	0 - 9

5.3.3. Map Descriptor Example.

```
// Server Side Map Descriptors

Map Descriptors
Map_Descriptor_Name,   Data_Array_Name,   Data_Array_Offset,   Function,   Node_name,   Inet_point_type,   Address,   Length
MR00_inputs_0,        An_inputs_0,        0,                  Passive,    MR_00,        analog_input,      0,         1
MR00_inputs_1,        An_inputs_0,        0,                  Passive,    MR_00,        analog_output,     1,         1
```

This can be any name but each name must be unique. Name will appear in FieldServer Map Descriptor status information screens.

The Data_Array_Name must be one found under Data_Arrays. Data will be sent from this Array to the device polling this MR.

This value specifies the offset into the Data Array where the data will be stored for testing with the scripts

Function may not be read or write since it implements a server. Function may only be passive.

Node name must be one found under Nodes, Node_Name. This defines this MR's station address.

The type of I/NET data this Map Descriptor points to

The I/Net starting point number.

The length or number of I/Net points this Map Descriptor points to.

Appendix A. Driver Notes

Appendix A.1. Scaling

The Server can scale point values when replying to a client poll and before writing them to data arrays. When a point value is retrieved from a server data array with scaling on the map descriptor, the value is normalized against the data array low and high scale values and then multiplied by the delta on the node scale values before being sent out in the reply to the client

Example:

Value retrieved from data array = 1

Data array scales at 0, 100

Value is scaled to $1/100 = 0.01$

Node scales at 0, 1000

Value is scaled to $0.01 * 1000 = 10$ and is sent to the requesting client.

When a point value is stored by the above map descriptor the reverse happens (so the value should be scaled correctly before being written to the server):

The incoming value = 10

Node scales at 0, 1000

Value is scaled to $10/1000 = 0.01$

Data array scales at 0, 100

Value is scaled to $0.01 * 100 = 1$ and is written to the data array.

Appendix A.2. I/Net Command Table

Command Code	Command Description	Implementation
00	Read All Points	Full Support
01	Read All Points (Power Failure)	Full Support
02	Control Output Points	Full Support
03	Control Input Points	Canned Response
04	Set Date & Time	Canned Response
06	Cold Start	Canned Response
XF	Diagnostics	No Response
13	Download Point Record	Partial
19	Download DDC Module	Canned Response
1A	Download Permanent Conversion Coefficient	Canned Response
1B	Download Stat Parameters	Canned Response
1C	Download ATS Parameters	Canned Response
23	Upload Point Record	Canned Response
28	Read DDC Line	Canned Response
29	Upload DDC Module	Canned Response
2A	Upload Permanent Conversion Coefficients	No Response
2B	Upload Stat Parameters	No Response
2C	Upload ATS Parameters	No Response
30	Read all DDC Lines	Canned Response
	Quick Poll (No Change)	PP
	Quick Poll (Item Changed)	PP & Changes

Appendix A.3. Interpreting Server Driver Error Statistics

The following error statistics can be found on the I-Net Server driver's Connection Error Statistics page using RuiNet.

Stat	Meaning	Solution
Protocol Errors	Unknown commands, undefined point types and other messages not conforming to the I-Net protocol.	Make sure correct hardware is connected, check wiring.
Bad Node	The driver could not find a map descriptor to store or retrieve point data.	Insert map descriptors for each point type defined on the node.
Data Object offline	Server Data is not valid.	Check client side of FieldServer for valid communications.

THIS PAGE INTENTIONALLY LEFT BLANK