



A Sierra Monitor Company

Driver Manual
(Supplement to the FieldServer Instruction Manual)
FS-8700-26 Notifier AFP 200/300/400

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after May 1, 2001

Driver Version:	1.02
Document Revision:	6

TABLE OF CONTENTS

1. Notifier AFP200/300/400 Description.....	3
2. Driver Scope of Supply.....	4
2.1. Supplied by FieldServer Technologies for this Driver.....	4
2.2. Provided by Supplier of 3 rd Party Equipment.....	4
3. Hardware Connections – AFP200.....	5
4. Configuring the FieldServer as a Notifier AFP200 Client.....	6
4.1. Data Arrays.....	6
4.2. Client Side Connection Descriptors.....	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>	<i>8</i>
4.4.2. <i>Driver Related Map Descriptor Parameters.....</i>	<i>8</i>
4.4.3. <i>Timing Parameters.....</i>	<i>8</i>
4.4.4. <i>Map Descriptor Example:.....</i>	<i>9</i>
5. Hardware Connections – AFP300/400.....	10
6. Configuring the FieldServer as a Notifier AFP300/400 Client.....	11
6.1. Data Arrays.....	11
6.2. Client Side Connection Descriptors.....	12
6.3. Client Side Node Descriptors.....	12
6.4. Client Side Map Descriptors.....	13
6.4.1. <i>FieldServer Related Map Descriptor Parameters.....</i>	<i>13</i>
6.4.2. <i>Driver Related Map Descriptor Parameters.....</i>	<i>13</i>
6.4.3. <i>Timing Parameters.....</i>	<i>13</i>
6.4.4. <i>Map Descriptor Example 1.....</i>	<i>14</i>
6.4.5. <i>Map Descriptor Example 2 – Store Trouble Status of System.....</i>	<i>14</i>
Appendix A. Special Driver Parameters.....	15
Appendix A.1. Notifier Data Types.....	15
Appendix A.2. Permissible Addresses.....	16
Appendix A.3. Zones.....	17
Appendix B. Driver Notes.....	18
Appendix B.1. Using Log Files to Test the Driver.....	18
Appendix C. Troubleshooting.....	20
Appendix D. Driver Error Messages.....	21

1. Notifier AFP200/300/400 Description

The Notifier AFP200/300/400 driver allows the FieldServer to transfer data to and from the Notifier AFP200, AFP300 and AFP400 Automatic Fire Alarm Panels over RS-232 using Notifier AFP200/300/400 protocol. The panel MUST output messages in English. The FieldServer functions as a Client with this driver.

The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer. As the AFP200 panel interface differs in many aspects from the AFP300/AFP400 interface, these panel interfaces will be discussed separately.

2. Driver Scope of Supply

2.1. Supplied by FieldServer Technologies for this Driver

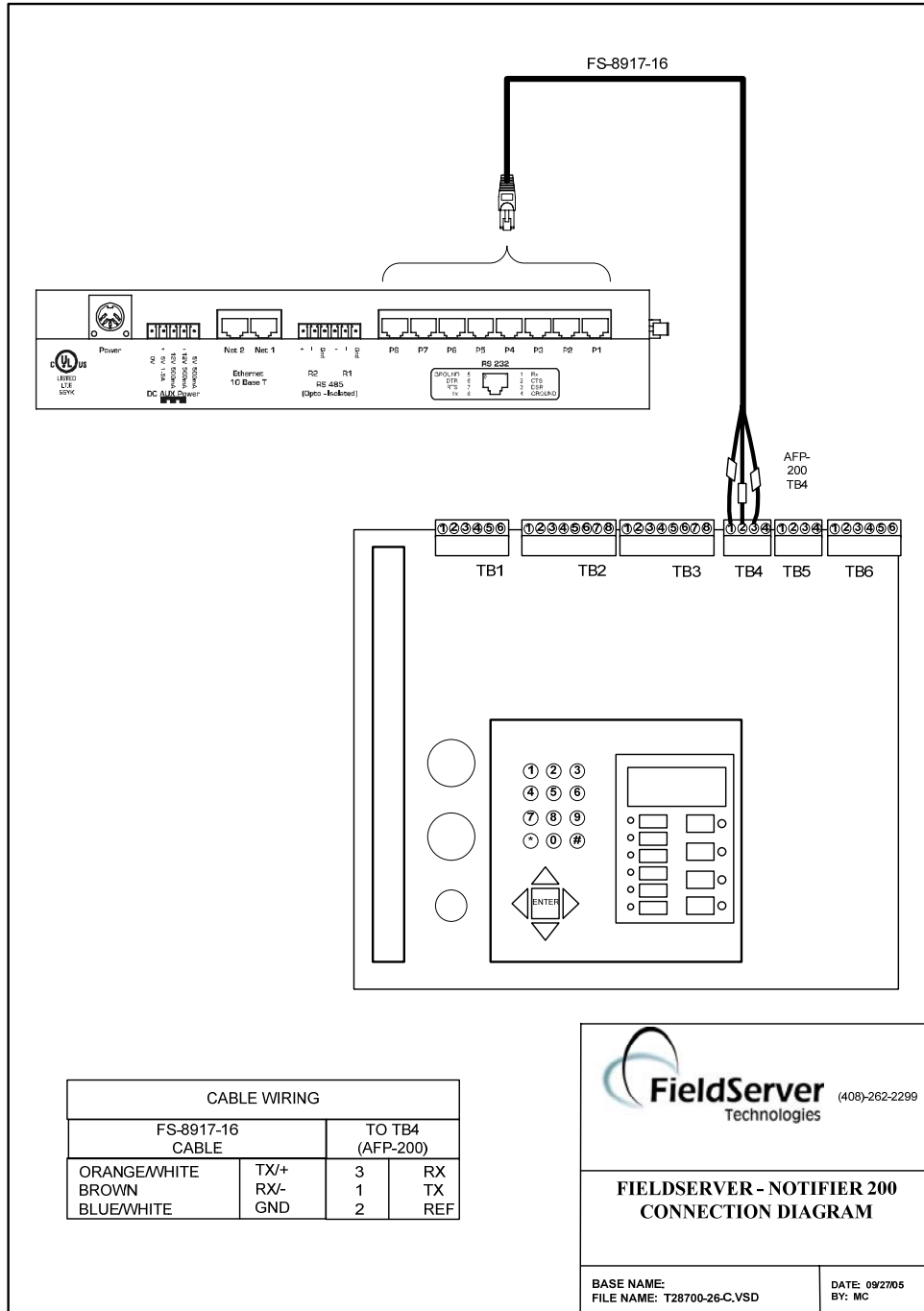
FieldServer Technologies PART #	DESCRIPTION
FS-8917-02	RJ45 to DB9F connector adapter
FS-8917-01	RJ45 to DB25M connector adapter
FS-59132	RS-485 connector adapter
FS-8917-16	RJ45 to DB25F connector adapter for use between the FieldServer and an Isolator (see following drawing)

2.2. Provided by Supplier of 3rd Party Equipment

PART #	DESCRIPTION
	Notifier APF200, AFP300 or AFP400 Fire Panel

3. Hardware Connections – AFP200

It is possible to connect a Notifier AFP200 device to any serial port. These ports just need to be configured for Notifier AFP200 in the configuration file.



4. Configuring the FieldServer as a Notifier AFP200 Client

The Notifier driver is both active and passive in that it will continuously scan for status' requested in the Client side Map Descriptor section of the configuration file (active element). In addition, it will receive all unsolicited messages sent by the panel to the FieldServer (passive element).

Passive Elements (unsolicited panel messages) usually constitute alarms and other time sensitive signals. These signals are acquired by the FieldServer on an interrupt basis and as such, are available immediately in the FieldServer Data Arrays.

Active elements (polled messages) usually constitute panel and point status updates. Due to the fixed baud rate of the communication channel, polled points are updated relatively slowly (about 2 seconds per point). Polled messages are considered to be a background activity, and generally this function is used to ensure that the database of the FieldServer (Data Arrays) remains consistent with the Notifier Database.

All Data Array elements are reset when the system reset button on the Notifier panel is pressed. In addition, certain specific Data Array elements will be reset when their related "Normal/Inactive" message is received.

4.1. Data Arrays

Data Arrays constitute the storage area for data in the FieldServer. Legal values for the available parameters are as follows:

Section Title		
Data_Arrays		
Column Title	Function	Legal Values
Data_Array_Name	Provide name for Data Array	Up to 15 alphanumeric characters
Data_Format	Provides data format	BIT; PACKED_BIT
Data_Array_Length	Number of Data Objects	1-10000

Example:

// Data Arrays		
Data_Arrays		
Data_Array_Name,	Data_Format,	Data_Array_Length
Det_01,	Bit,	500
Mod_01,	Bit,	500
Det_02,	Bit,	500
Mod_02,	Bit,	500
DA_Zones,	Bit,	100
DA_System,	Bit,	16

4.2. Client Side Connection Descriptors

In this section of the configuration, the port (and its associated properties) being used for connection to the Notifier panel is defined.

Section Title		
Connections		
Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2 ¹
Baud	Specify baud rate	2400
Parity	Specify parity	Even
Data_Bits	Specify data bits	7
Stop_Bits	Specify number of stop bits	1
Protocol	Specify protocol	Notifier – AFP
Timeout	Specify maximum response time	>10 S
IC_Timeout	Specify inter character timeout	>1.0 S

Example:

```
// Client Side Connections
Connections
Port, Baud, Parity, Data_Bits, Stop_Bits, Protocol, Timeou,t, IC_Timeout
P8, 2400, Even, 7, 1, NOTIFIER-AFP, 10.0s, 1.0s
```

4.3. Client Side Node Descriptors

The FieldServer automatically assigns the Device Internal Node_ID of station 257.

Section Title		
Nodes		
Column Title	Function	Legal Values
Node_Name	Provide name for node	Up to 32 alphanumeric characters
Node_ID	Node ID of physical Server node	The Node_ID gets set to the Node_ID configured in the AFP panel.
Protocol	Specify protocol used	Notifier-AFP
Port	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2 ¹
Timeout	Specify Maximum Response Time	>10 S

Example:

```
Nodes
Node_Name, Node_ID, Protocol, Port, Timeout
AFP_1, 0, NOTIFIER-AFP, P8, 10.0s
```

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

Map Descriptors determine where in the Data Arrays the various points from the Notifier panel will be mapped (stored).

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, 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
Device_Type	Device Type in Panel	Detector, Module, Zone, Bell, System
Data_Type	Data type in Panel	Refer to Appendix A.1
Address	Starting address of read block	Refer to Appendix A.2
Length	Number of modules, alarms, etc.	1-99

4.4.3. Timing Parameters

Column Title	Function	Legal Values
Scan_Interval	Specify minimum Interval between Scans	> 10.0 S

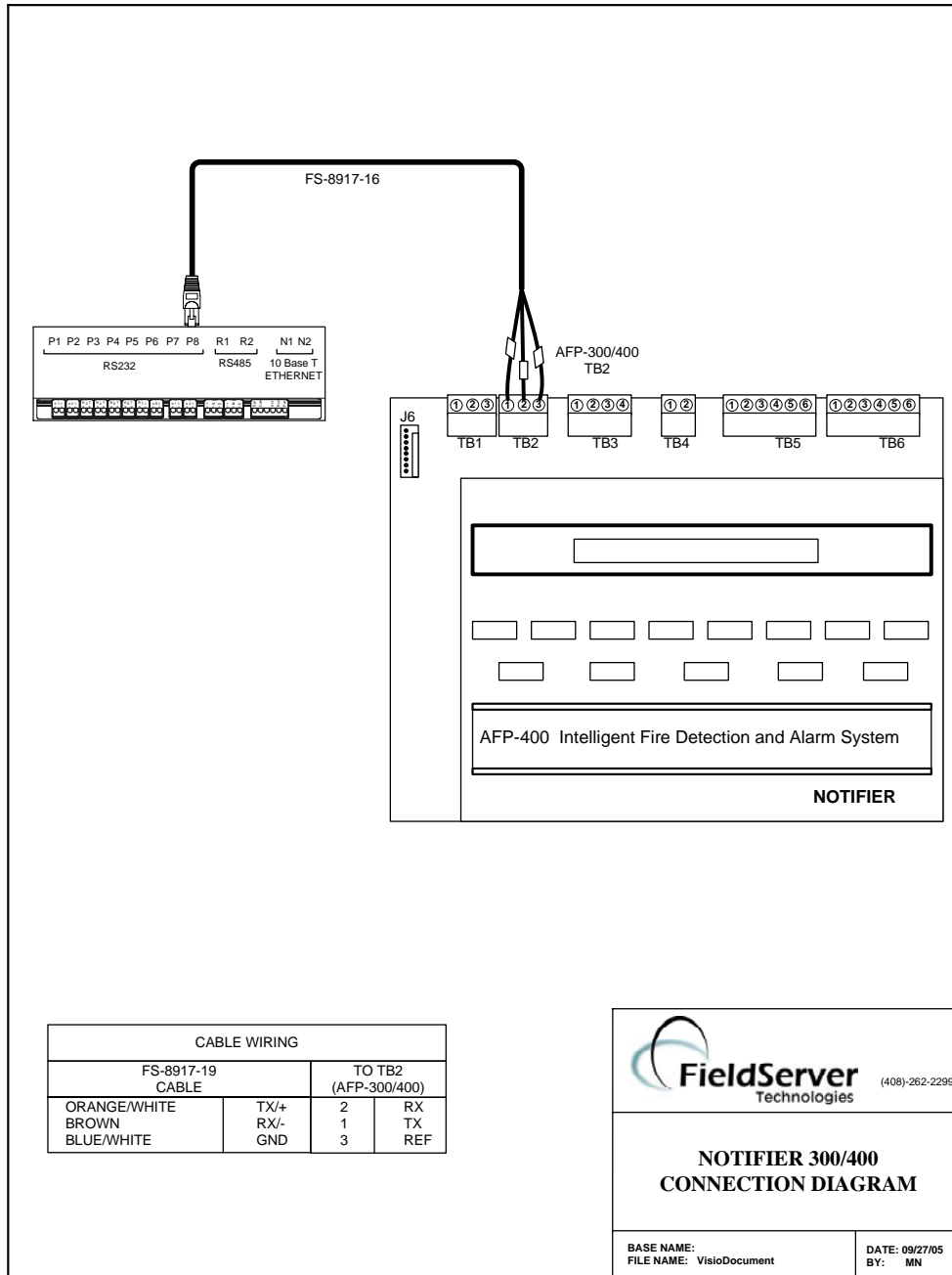
4.4.4. Map Descriptor Example:

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

Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	Function,	Device_Type,	Data_Type,	Node_Name,	Address,	Length,	Scan_Interval
AFP200_det_alm_01,	Det_01,	1,	Rdbc,	Detector,	Alarm,	AFP_1,	01,	99,	2
AFP200_det_trb_01,	Det_01,	101,	Passive,	Detector,	Trouble,	AFP_1,	01,	99,	
AFP200_det_pas_01,	Det_01,	201,	Passive,	Detector,	Pre_Alarm,	AFP_1,	01,	99,	
AFP200_det_nin_01,	Det_01,	301,	Passive,	Detector,	Not_Installed,	AFP_1,	01,	99,	
AFP200_mod_alm_01,	Mod_01,	1,	Rdbc,	Module,	Alarm,	AFP_1,	01,	99,	2
AFP200_mod_trb_01,	Mod_01,	101,	Passive,	Module,	Trouble,	AFP_1,	01,	99,	
AFP200_mod_pas_01,	Mod_01,	201,	Passive,	Module,	Pre_Alarm,	AFP_1,	01,	99,	
AFP200_mod_nin_01,	Mod_01,	301,	Passive,	Module,	Not_Installed,	AFP_1,	01,	99,	
AFP200_mod_nal_01,	Mod_01,	401,	Passive,	Module,	Non_Alarm,	AFP_1,	01,	99,	

5. Hardware Connections – AFP300/400

It is possible to connect a Notifier AFP300 or AFP400 device to any serial port. These ports just need to be configured for Notifier AFP300/400 in the configuration file.



6. Configuring the FieldServer as a Notifier AFP300/400 Client

The Notifier driver is both active and passive in that it will continuously scan for status' requested in the Client side Map Descriptor section of the configuration file (active element). In addition, it will receive all unsolicited messages sent by the panel to the FieldServer (passive element).

Passive Elements (unsolicited panel messages) usually constitute alarms and other time sensitive signals. These signals are acquired by the FieldServer on an interrupt basis and as such, are available immediately in the FieldServer Data Arrays.

Active elements (polled messages) usually constitute panel and point status updates. Due to the fixed baud rate of the communication channel, polled points are updated relatively slowly (about 8 seconds per point). Therefore, if the status of a large number of points is desired, it could mean that it could take minutes for the status of a point to update in the Data Array, e.g. an alarm point that appeared immediately in the data array due to an unsolicited message from the panel would only clear when the point's status is updated and the alarm has cleared in the panel. As the alarm was reported immediately, this does not compromise the integrity of the system.

6.1. Data Arrays

Data Arrays constitute the storage area for data in the FieldServer. Legal values for the available parameters are as follows:

Section Title	Function	Legal Values
Data_Arrays		
Column Title	Function	Legal Values
Data_Array_Name	Provide name for Data Array	Up to 15 alphanumeric characters
Data_Format	Provides data format	BIT; PACKED_BIT
Data_Array_Length	Number of Data Objects	1-10000

Example:

```
// Data Arrays

Data_Arrays
Data_Array_Name,      Data_Format,      Data_Array_Length
Det_01,              Bit,              500
Mod_01,              Bit,              500
Det_02,              Bit,              500
Mod_02,              Bit,              500
DA_Zones,            Bit,              100
DA_System,           Bit,              1
```

6.2. Client Side Connection Descriptors

In this section of the configuration, the port (and its associated properties) being used for connection to the Notifier panel is defined.

Section Title		
Connections		
Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2 ²
Baud	Specify baud rate	2400
Parity	Specify parity	None
Data_Bits	Specify data bits	8
Stop_Bits	Specify number of stop bits	1
Protocol	Specify protocol	Notifier – AFP
Timeout	Specify maximum response time	>10s
IC_Timeout	Specify inter character timeout	>1.0s

Example:

```
// Client Side Connections
Connections
Port, Baud, Parity, Data_Bits, Stop_Bits, Protocol, Timeou,t, IC_Timeout
P8, 2400, None, 8, 1, NOTIFIER-AFP, 10.0s, 1.0s
```

6.3. Client Side Node Descriptors

The FieldServer automatically assigns the Device Internal Node_ID of station 257.

Section Title		
Nodes		
Column Title	Function	Legal Values
Node_Name	Provide name for node	Up to 32 alphanumeric characters
Node_ID	Node ID of physical Server node	The Node_ID gets set to the Node_ID configured in the AFP panel.
Protocol	Specify protocol used	Notifier-AFP
Port	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2 ²
Timeout	Specify Maximum Response Time	>10 S

Example:

```
Nodes
Node_Name, Node_ID, Protocol, Port, Timeout
AFP_1, 0, NOTIFIER-AFP, P8, 10.0s
```

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.

6.4. Client Side Map Descriptors

Map Descriptors determine where in the Data Arrays the various points from the Notifier panel will be mapped (stored).

6.4.1. 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, Passive

6.4.2. 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
Device_Type	Device Type in Panel	Detector, Module, Zone, Bell, System
Data_Type	Data type in Panel	Refer to Appendix A.1
Address	Starting address of read block	Refer to Appendix A.2
Length	Number of modules, alarms, etc.	1-99

6.4.3. Timing Parameters

Column Title	Function	Legal Values
Scan_Interval	Specify minimum Interval between Scans	> 10.0 S

6.4.4. Map Descriptor Example 1

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

Map Descriptors									
Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	Function,	Device_Type,	Data_Type,	Node_Name,	Address,	Length,	Scan_Interval
AFP200_det_alm_01,	Det_01,	1,	Rdbc,	Detector,	Alarm,	AFP_1,	101,	99,	2
AFP200_det_trb_01,	Det_01,	101,	Passive,	Detector,	Trouble,	AFP_1,	101,	99,	
AFP200_det_pas_01,	Det_01,	201,	Passive,	Detector,	Pre_Alarm,	AFP_1,	101,	99,	
AFP200_det_nin_01,	Det_01,	301,	Passive,	Detector,	Not_Installed.	AFP_1,	101,	99,	
AFP200_mod_alm_01,	Mod_01,	1,	Rdbc,	Module,	Alarm,	AFP_1,	101,	99,	2
AFP200_mod_trb_01,	Mod_01,	101,	Passive,	Module,	Trouble,	AFP_1,	101,	99,	
AFP200_mod_pas_01,	Mod_01,	201,	Passive,	Module,	Pre_Alarm,	AFP_1,	101,	99,	
AFP200_mod_nin_01,	Mod_01,	301,	Passive,	Module,	Not_Installed,	AFP_1,	101,	99,	
AFP200_mod_nal_01,	Mod_01,	401,	Passive,	Module,	Non_Alarm,	AFP_1,	101,	99,	

6.4.5. Map Descriptor Example 2 – Store Trouble Status of System

This Map Descriptor will store the Trouble Status of the entire system. The driver looks for "TROUBL IN SYSTEM" in a message from the panel and stores a 1 at the determined offset to indicate that there is a trouble in the system. Since this is a discrete point, a Data Array length of 1 is sufficient. The Data Array will be reset on "SYSTEM RESET" or "ALL SYSTEMS NORMAL".

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

Map Descriptors									
Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	Function,	Device_Type,	Data_Type,	Node_Name,	Address,	Length,	Scan_Interval
AFP200_det_trb_01,	DA_System,	101,	Passive,	system,	Trouble,	AFP_1,	-,	99,	

Appendix A. Special Driver Parameters

Appendix A.1. Notifier Data Types

It is possible to scan separately for the various types of data points in the Notifier panel. In particular, modules have several data types known as “type codes.” (Refer to the relevant Operations Manual, for a detailed description of the behavior of each of the type codes).

Permissible parameters for the Data_Type field in the Notifier Map Descriptor are as follows:

Alarm	Pre_Alarm
Trouble	Not_Installed
Non_Alarm	Supervisory
Tamper	Non_Fire
Hazard Alert	Fire_Control
Abort_Switch	Silence
Evacuate	PAS_Inhibit
Second_Shot	Security

These data types relate to the type codes as follows:

Alarm Data Type

This data type is valid for the following Device Types:

- Detector
- Module
- Zone
- Bell

A Map Descriptor with this data type declared will set a bit in the designated data array at the designated offset if any one of the following type codes are assigned to the related Notifier SLC address in the Notifier Panel and an alarm is active at that point. Alarms are reset by pressing System Reset on the Notifier Panel.

Monitor	Pull Station
Smoke Detect	Heat Detect
Blank	Waterflow
Man. Release	Trouble Man.
Tele Page	All call page
Man. Rel. Delay	Sprinkler Sys
Comb. Monitor	All Detector Types

Trouble Data Type

This data type is valid for the following device types:

- Detector
- Module
- System

A Map Descriptor with this data type declared will set a bit in the designated data array at the designated offset if the related SLC network device goes into trouble status. Troubles are reset by pressing System Reset on the Notifier Panel. See example in section 6.4.5.

Pre Alarm Data Type

This data type is valid for the following device types:

- Detector
- Module

A Map Descriptor with this data type declared will set a bit in the designated data array at the designated offset if the pre-alarm for a device becomes active. Pre_Alarms are reset by pressing System Reset on the Notifier Panel.

Not Installed Data Type

This Data Type is valid for the following device types:

- Detector
- Module

A Map Descriptor with this data type declared will set a bit in the designated data array at the designated offset if there is no device installed at the related SLC network address. The “Not_Installed” status of point is continuously updated by the panel.

Non Alarm Data Type

This data type is valid for the following device types:

- Modules

A Map Descriptor with this data type declared will set a bit in the designated data array at the designated offset if any one of the following type codes are assigned to the related SLC address in the Notifier Panel, and the status of that address is active:

Supervisory	Tamper
Non_Fire	Hazard_Alert
Fire_Control	Abort_Switch
Silence	System_Reset
Evacuate	PAS_Inhibit
Second_Shot	Security

It is also possible to use each of these type codes as a data type, in which case only the related type code will activate a bit in the data array. Depending on the nature of the non_alarm type code, active status' are reset either by inactive status reported, or by System Reset being pressed, or both.

Appendix A.2. Permissible Addresses

Permissible addresses for the various device types are as follows

Detectors:	1-99
Modules:	1-99
Zones:	1-99
Bells:	1-4
System:	--

Appendix A.3. Zones

If it is necessary for the related zone to be scanned by the FieldServer, then the Zone must have its default text programmed in the Notifier Panel, i.e. "Zone xx", where xx is the zone number. A Map Descriptor must also exist to map the required zones to a designated data Array.

Active zones are reset only when the system reset button is pressed on the Notifier panel.

Note that the Notifier protocol only reports the first of the five possible zones for any point in the FieldServer. It is therefore advisable to assign the five zones for every point with this in mind

Appendix B. Driver Notes

Appendix B.1. Using Log Files to Test the Driver

These notes are intended for FieldServer tech support only.

It is possible to construct a test script using an ASCII file to send messages to the driver. This feature was added in version 1.02a of the driver.

This is an example of a log file. You can see the file is an ASCII file. Lines that begin `//` are ignored.

```
// -----  
// Comments begin with a double slash  
//  
// Specify the file name in the Client mapdesc using the "Log_File_Name" parameter  
// Set the address equal to the 1st line of the file to be sent  
// Set the length equal to the number of lines to be sent  
// when the final line is sent then no more messages are sent  
// The driver removes up to 2 cr's and 2 lf's chars looking backward from the end of the line  
// and then appends one crlf pair.  
// Except when the 1st byte of the line's ascii value is less than 32  
~A  
1  
D005
```

The following example illustrates the usage.

Map_Descriptor_Name, SendLogFileMsgs ,	Data_Array_Name, DA_AI_01 ,	Data_Array_Offset, 0 ,	Function, wrbc ,	Node_name, DEV2 ,	Address, 1,	Length, 50 ,	scan_interval, 0.2s ,	Log_File_Name s2681a.inj
---	--------------------------------	---------------------------	---------------------	----------------------	----------------	-----------------	--------------------------	-----------------------------

First line of file that will be sent.

In this example the 1st line to be sent will be line 1 because the address equals 1.

The name of the file.

How many lines will be sent.

Appendix C. Troubleshooting

- If the FieldServer reboots when connected to the Panel Serial port, then it is most likely that an Optical Isolator is required to balance ground potential differences. Such differences have been known to damage the FieldServer serial port, and therefore it is recommended that this action is taken as soon as the symptom is observed.

Appendix D. Driver Error Messages

When a message is marked with a '*' this means that when the same error occurs again the message will be suppressed. This is done to stop the error log being filled with duplicate messages which do not convey additional information.

Message	Explanation
NAFP:#01 Err. Need an MD to store detector msg. Detector=102 Device_Type=101 Data_Type=Not_Installed	The message is printed when the driver is trying to store information about a detector and cannot find a Map Descriptor to use for the storage. The message reports the detector number, the device type (see message #02 for device types) and the Data_Type. This information is sufficient to create a new MapDesc to define a storage location. In the example the Client polled for detector data - the panel responded that the detector wasn't installed and no MD was defined to store 'Not_Installed' responses. Each time one of these error's occurs the driver increments the MSG_IGNORED stat. This stat does not increment the error count on the connection overview page of RUINET. ³
NAFP:#02 FYI. Device_Types: Det=101 Mod=102 Zone=103 Bell=104 Sys=105	This message is for your information only and does not require corrective action on your part.
NAFP:#03* Err. Unknown Type. Expected D/M/Z/B. Rcvd=%c=0x%x	There is no corrective action you can take to eliminate this error. If the error occurs rarely then it may be the result of a corrupted message – this can occur from noise on the communication line. If it occurs repeatedly then you need to take a log and report the problem to tech Support. A procedure for taking a log is provided in the FieldServer Trouble Shooting Guide. The error occurs when parsing a message.
NAFP:#04 FYI. Emulating 2003 Firmware.	This message is for your information only and may be safely ignored. The message reports that the Server side of the driver is emulating the new (summer 2003) panel firmware.
NAFP:#05a Err. MD too short. No response will be sent. Rqd Offset=%d Actual Range=%d to %d	The Server side of the driver produces this message if the address range of a Server MD is too short to serve a response for the device/module etc. requested. ³
NAFP:#06 Err. Cant Store. RqdOffset=%d DA=%s Length=%d	The offset into the data array required to store the data is larger than the number of elements in the data array. Adjust the length of the reported Data Array (DA). There are a few variations of this message which are used by tech support to trace the source of the problem. ³
NAFP:#08a Err. Cant Get. RqdOffset=%d DA=%s Length=%d	

³ Modify the CSV file, download the modified file to the FieldServer and reset the FieldServer for the changes to take effect.

THIS PAGE INTENTIONALLY LEFT BLANK.