MSA The Safety Company | **fieldserver**

Driver Manual

FS-8700-52 Notifier Italia AM6000

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

MSAsafety.com

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

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

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

Contents

1   Description

The Notifier Italia AM6000 driver allows the FieldServer to transfer data to and from devices over either RS-232 or RS-485 using Notifier Italia Am6000 protocol. There are eight RS-232 and two RS-485 on each FieldServer. The Notifier Am6000 serial driver is a passive client only driver. This means that the driver is capable of processing and storing data from unsolicited messages sent from a Notifier panel. It is not capable of polling for data or sending data to a panel. As the driver does not provide server side functionality it is not capable of emulating a Notifier panel.

The driver receives messages intended for a system printer and interprets these messages by filling in data arrays in the FieldServer. This data is available for other devices or PLC's to read.

As the messages do not contain information which would allow this driver to differentiate messages from one panel or another, no more than one panel can be connected to a port.

2   Driver Scope of Supply

2.1   Supplied by MSA Safety

Part #	Description
FS-8915-10	UTP cable (7 foot) for Ethernet connection
FS-8915-10	UTP cable (7 foot) for RS-232 use
FS-8917-02	RJ45 to DB9F connector adapter
FS-8917-01	RJ45 to DB25M connection adapter
	RS-485 connection adapter
FS-8700-52	Driver Manual

3   Hardware Connections

The FieldServer is connected to the Notifier Italia AM6000 as shown in connection drawing. Configure the Notifier Italia AM6000 according to manufacturer's instructions.
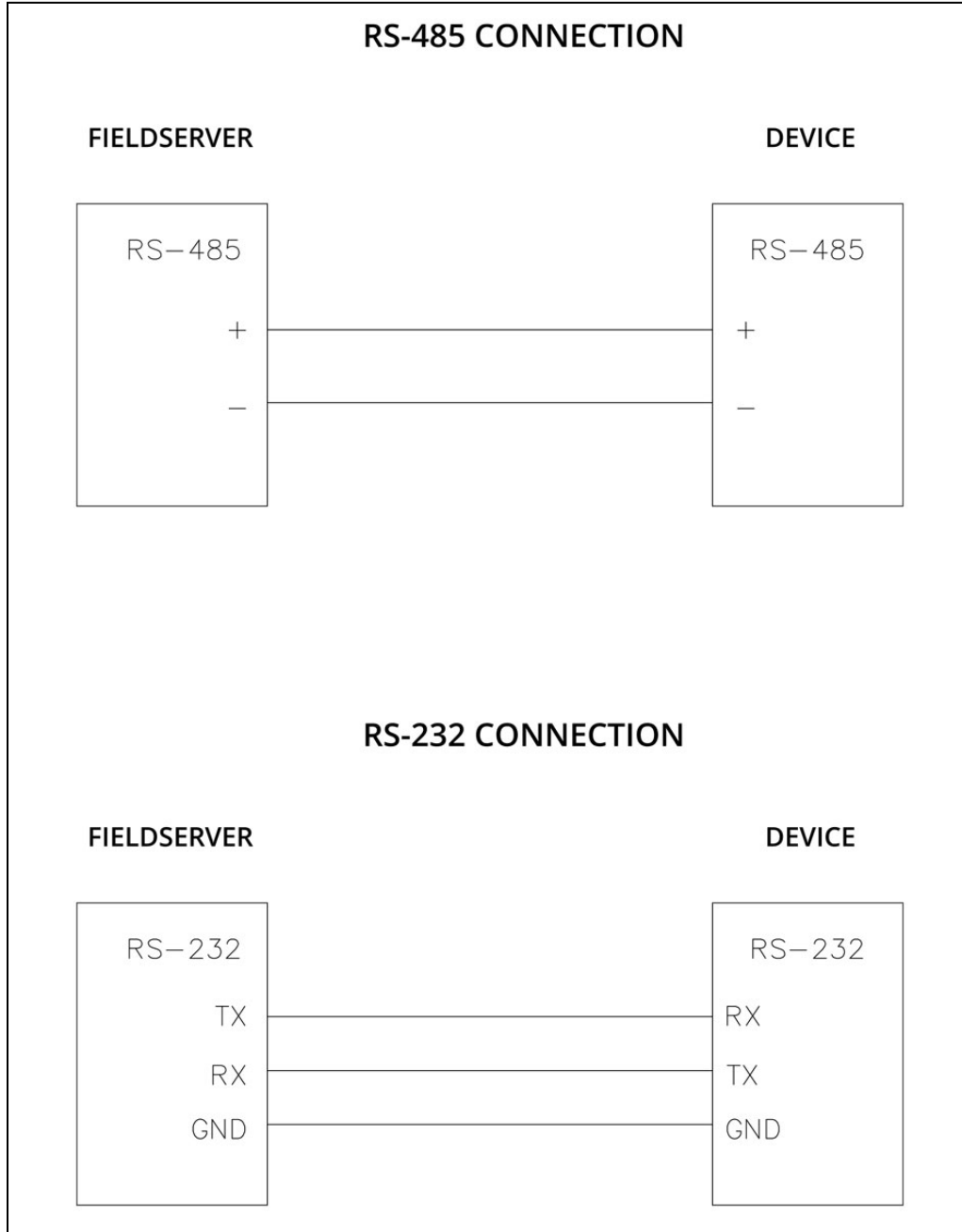
4   Data Array Parameters

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

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

Example

```
//   Data Arrays
Data_Arrays

Data_Array_Name    , Data_Format    , Data_Array_Length
DA_AI_01           , UInt16         , 200
DA_AO_01           , UInt16         , 200
DA_DI_01           , Bit            , 200
DA_DO_01           , Bit            , 200
```

5   Client Side Configuration

For detailed information on FieldServer configuration, 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 FS).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a Notifier Italia AM6000 Server.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for Notifier Italia AM6000 communications, the driver independent FieldServer buffers need to be declared in the "Data Arrays" section, the destination device addresses need to be declared in the "Client Side Nodes" section, and the data required from the servers needs to be mapped in the "Client Side Map Descriptors" section. Details on how to do this can be found below.

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

5.1   Client Side Connection Parameters

Section Title		
Connections		
Column Title	**Function**	**Legal Values**
Port	Specify which port the device is connected to the FieldServer.	P1-P2, R1-R2[1]
Protocol	Specify protocol used.	AM6000
Baud*	Specify baud rate.	2400 (only baud rate supported by the Notifier port)
Parity*	Specify parity.	Even
Data_Bits*	Specify data bits.	7
Stop_Bits*	Specify stop bits.	1
Poll_Delay*	Time between internal polls.	0-32000 seconds, 1 s

Example

```
//   Client Side Connections
Connections
Port    , Protocol    , Baud     , Parity    , Poll_Delay
P1      , AM6000     , 24000    , Even      , 0.100 s
```

5.2   Client Side Node Descriptors

Section Title		
Nodes		
Column Title	**Function**	**Legal Values**
Node_Name	Provide name for node.	Up to 32 alphanumeric characters
Node_ID	This commonly used parameter is not required for this driver.	
Protocol	Specify Protocol used.	AM6000
Port	Specify through which port the device is connected to the FieldServer.	P1-P2, R1-R2[1]

Example

```
//   Client Side Nodes
Nodes
Node_Name    , Protocol    , Port
Panel1          , AM6000    , P1
```

[1] Not all ports shown may be supported by the hardware. Consult the appropriate Instruction manual for details of the 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 "Data Array" section above
Data_Array_Offset	Starting location in Data Array.	0 to maximum specified in "Data Array" section above
Function	Function of Map Descriptor.	Passive

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 "Client Node Descriptor" above
Data_Type	This commonly used parameter is not used or required by this driver.	
Length	Length of Map Descriptor – this value is important when a "Block Ack" message is received as it tells the driver how much of the array to set to zero.	1-1000
Address	Loop/Device Address. In the format mmnn where mm is the loop number and nn is the device number. Together they form a composite device number.	501, 302
AM6000 Driver Specific Parameters & Keywords		
Am6000_TypeID	Used to differentiate between device and system type messages.	Device-only, System_trbl, system-Styl2, catch-all; Additional notes are provided in **Section 8.1.1**
Am6000_Status	Allows direction of alarm or trouble states to different data arrays.	Any, Alarm, trouble, erase, on, erase on; Additional notes are provided in **Section 8.1.2**
AM6000_Ackable	Tells the driver that the data area of this Map Descriptor must be set back to zero when a block Ack is received.	Additional notes are provided in **Section 8.1.3**
Am6000_Simulation	Provided for debugging and test purposes only. Tells the driver which debug message to send. This parameter is for use by MSA Safety only.	Additional notes are provided in **Section 8.1.4**

5.3.3 Timing Parameters

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

5.4   Map Descriptor Examples

5.4.1   Example 1

Section Title Map Descriptors Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Function	Node_name	Address	Length
DEVICES_100_199	DEVICE_STATES_LOOP1	0	Passive	Node_A	500	100

, Scan_Interval	, AM600_TypeID	, AM600_Ackable
, 1.0s	, Device-only	, Yes

In the above example:

- Data_Array_Name – Name of the Data Array that will be updated when this Map Descriptor is used.

- Data_Array_Offset – The position in the Data Array that corresponds to the device whose number is equal to the address parameter. In this example, device 500's state will be stored in the 0th element of the data array called DEVICE_STATES_LOOP1.

- Function – All Map Descriptors for the AM6000 Notifier driver must be passive.

- Node_Name – Connects this Map Descriptor to a node whose name is "Node_A" and thus connects the Map Descriptor to a port.

- Address – The device address of the first device to be processed by this Map Descriptor. Devices are allocated to loops. The device address consists of a mmnn format where mm is the loop number and nn is the device number. In this example, the address is set to 500 – the first device to be processed is device 0 on loop 5.

- Length – The range of device address is determined by this parameter. In this example, all devices 500 to 599 will be processed using this Map Descriptor.

- AM600_TypeID – This keyword indicates that this Map Descriptor will only be used to process device state messages. This makes the address and length parameters very important.

- AM600_Ackable – When a Block Acknowledge or System Reset message is received the driver will set the portion of the Data Array DEVICE_STATES_LOOP1 starting at Data_Array_Offset for length number of elements to zero.

5.4.2 Example 2

Section Title					
Map Descriptors					
Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Function	Node_name	Length
MD1	SYSTEM_TROUBLES2	5	Passive	Node_A	50

| , Scan_Interval | , AM600_TypeID | , AM600_Ackable |
| , 1.0s | , system-styl2 | , Yes |

In the above example:

- Data_Array_Name – Name of the Data Array that will be updated when this Map Descriptor is used.

- Data_Array_Offset – The position in the Data Array that corresponds to the device whose number is equal to the address parameter.

- Function – All Map Descriptors for the AM6000 Notifier driver must be passive.

- Node_Name – Connects this Map Descriptor to a node whose name is "Node_A" and thus connects the Map Descriptor to a port.

- AM600_TypeID – System Trouble Messages are processed using this Map Descriptor. Only one such Map Descriptor is necessary for the whole system. This indicates that Style2 is being used – thus when system trouble message number **n** is processed, the nth element of data array starting at location 5 will be set to 1.

- AM600_Ackable – When a Block Acknowledge or System Reset message is received the driver will set the portion of the Data Array DEVICE_STATES_LOOP1 starting at Data_Array_Offset for length number of elements to zero.

6 Server Side Configuration

The Notifier AM6000 driver cannot be configured to act as a server. Some server functionality has been developed to provide automated testing for the client functionality; It is undocumented and unsupported. It is possible to have this functionality extended and documented, typically at additional cost.

7   Advanced Topics

7.1   Support for Panels Operating in Italian

Some panels can be configured to send messages in English or Italian. The driver is capable of parsing some messages when the panel is in Italian Mode.

The following list illustrates the types of messages that can be successfully parsed:

- **Alarm message**. Updates *device-only* map descriptors when the *AM600_status* keyword is set to *Any* or *Alarm.*

    ALLARME : PULSANTE MAN PULSANTE MAGAZZINO   09:54 22/10/02 101

- **Trouble message**. Updates device-only map descriptors when the AM600_status keyword is set to Any or Trouble.

    GUASTO : OTTICO ANAL. MAGAZZINO   09:55 22/10/02 101

- **Block Acknowledge message**.

    ******** RICONOSCIMENTO EVENTI ********   09:54 22/10/02

- **System Reset message**.

    RESET SISTEMA   ATTIVATO   09:55 22/10/02

Only map descriptors with the following *Am6000_typeID = Device-only* will update correctly. And this will only be the case when the *AM6000_*Status parameter is set to *Any, Alarm, Trouble*

7.1.1   Limitations

Panel status messages cannot be processed when the panel is operating in Italian Mode. You can expect that map descriptors with the following *Am6000_TypeID* keywords will **not** be update correctly.

8   Driver Notes

8.1   Map Descriptor Keywords

8.1.1   AM6000_TypeID

This keyword sets up a category of messages. If you are interested in device state's use the keyword *device-only*. If you are interested in system trouble messages use the keyword *system-trbl* (an alternate method for reporting system trouble messages is provided using the keyword *system-styl2*.)

Keyword	Function
device-only	Only device based messages are processed using map descriptors with this keyword. The **Address** and **Length** parameters define the range of devices processed using the map descriptor. A device address consists of a loop and device number. This driver treats the two parts as a single device address. Thus loop 5 device 2 becomes address 502 for this driver and a map descriptor with an address of 500 and a length of 100 would process a message from device 2 on loop 5.
system-trbl	*System Trouble* messages are processed a numeric value is stored in the first element of the map descriptor's data array. If a new system trouble message is processed, then the value is overwritten with the new value.
system-styl2	This is an alternate style for processing *System Trouble* messages. If a system trouble message is received, then it is matched against the list. The array location is determined by the index value form the table and that location is set to 1. Make sure the length parameter is set to at least 50.
Catch-all	The use of this keyword is described in **Section 8.3.2**.

8.1.2   AM6000_Status

Device state messages report a number of different states for the device. These states are referred to as the device status.

The following are possible values of this parameter. Any, alarm, trouble, on, erase, erase on.

If you do not use this parameter in a *device-only* map descriptor then the driver uses the default value of "Any". This means that any device state message will result in the same data array being updated. Thus, a trouble/on/alarm message will result in the array being set to 1.

Use this parameter to maintain separate arrays for each state. For one device address range you would have multiple map descriptors, each with a different AM6000_Status keyword.

Note that the keyword, *erase*, results in the data array value being set to zero when a message reports the device state as 'ERASE'. The keyword *erase on* sets the value to one, when the same message is received.

8.1.3   AM6000_Ackable

This parameter tells the driver that the data array portion associated with this map descriptor can be set to zero when a Block Acknowledge or System Reset message is received.

Set this parameter to one of the following legal keywords: *yes, no.*

The Data_Array_Location and Length parameters are used to determine what portion of the associated data array must be set to zero.

8.1.4   AM6000_Simulation

This keyword is for used by MSA Safety for testing this driver.

8.2 Statistics

8.2.1 Standard Stats

This driver does not keep statistics for each map descriptor. Statistics are maintained for the connection to the AM6000 Notifier device.

8.2.2   Exposing Driver Stats

In addition to the standard FieldServer communication statistics described in the FieldServer User's Manual, the Notifier AM6000 Driver can also expose some driver statistics by writing data to a data array. A special map descriptor is required. The driver recognizes the map descriptor by its name which must be "AM6000-stats".

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

```
Nodes
Node_Name          , Protocol
Stats_Node         , AM6000


Data_Arrays,
Data_Array_Name    , Data_Format         , Data_Array_Length
AM6000_STATS       , UINT16              , 500


Map_Descriptors
Map_block_Name     , Data_Array_Name    , Data_Array_Offset    , Function   , Node_name    , Address    , Length
AM6000-Stats       , AM6000_STATS       , 0                    , Passive    , Stats_Node   , 0          , 500
```

When the driver sees this map descriptor it uses the data array AM6000_STATS (in this example) to store driver specific statistics. Only one of these map descriptors may be specified per FieldServer.

The offset into the array is based on the port number. 30 arrays locations are used per port.

The offset is obtained by multiplying the port number by 30.

The driver stores the following data:

PORT						Description
0	1	2	3	…	8	
Array Offset						
0	50	100	150		400	Available for future use
1	51	101	151		401	Number of message's Received. Count increments each time a CR-LF is found.
2	52	102	152		402	Number of messages that the driver parsed successfully. This means that the message did not contain any major formatting errors that made it impossible for the message to be parsed correctly.
3	53	103	153		403	Number of times that a 'Block Ack' message has been received.
4	54	104	154		404	Number of times that a 'System Reset' message has been received.
5	55	105	155		405	Number of times that a 'Signal Silence' message has been received.
6	56	106	156		406	Number of ignored messages. Includes ignored status and ignored block messages.
7	57	107	157		407	Number of messages that driver ignored but which the driver thinks could contain important status information.
8	58	108	158		408	Number of messages that driver ignored but which the driver thinks could contain important status information.
9	59	109	159		409	Number of messages that driver ignored but which the driver thinks could contain important system information.

10	60	110	160		410	Available for future use
11	61	111	161		411	Available for future use
12	62	112	162		412	Available for future use
13	63	113	163		413	Available for future use
14	64	114	164		414	Available for future use
15	65	115	165		415	Available for future use
16	66	116	166		416	Available for future use
17	67	117	167		417	Available for future use
18	68	118	168		418	Available for future use
19	69	119	169		419	Available for future use
20	70	120	170		420	Available for future use
21	71	121	171		421	Available for future use
22	72	122	172		422	Available for future use
23	73	123	173		423	Available for future use
24	74	124	174		424	Available for future use
25	75	125	175		425	Available for future use
26	76	126	176		426	Available for future use
…	…	…	…		…	
49	99	149	199		449	Available for future use

8.3   Ignored Messages

Ignored messages are very important since they do not result in the data arrays being updated. Messages are ignored for one of **two reasons**.

- The driver might not understand the message or know what to do with the data. The message may contain a keyword or be formatted in a way that cannot be understood. MSA Safety needs to be informed when new features are added to the protocol by the manufacturer in order that the driver may be updated.

- You may have omitted to define a map descriptor which tells the driver what to do with the data from an incoming message. For example, say a map descriptor is defined for address 501 with a length of 50 but a message is received from device 575. Clearly the map descriptor's device address range does not extend as far as 575 and thus the driver doesn't know where to store device 575's data.

8.3.1   Viewing Ignored Messages

Ignored messages are sent to the User Messages Error page (E) or the User Messages Driver page (F) in FS-GUI. Remember that ignored messages are messages that the driver doesn't understand or doesn't know where to store.

The messages are identified as:

AM6000:#9	(sent to F Screen)
AM6000:#10	(sent to E Screen)
AM6000:#11	(sent to F Screen)

The driver will stop sending messages to these screens if you use the 'catch-all' method outlined in **Section 8.3.2**.
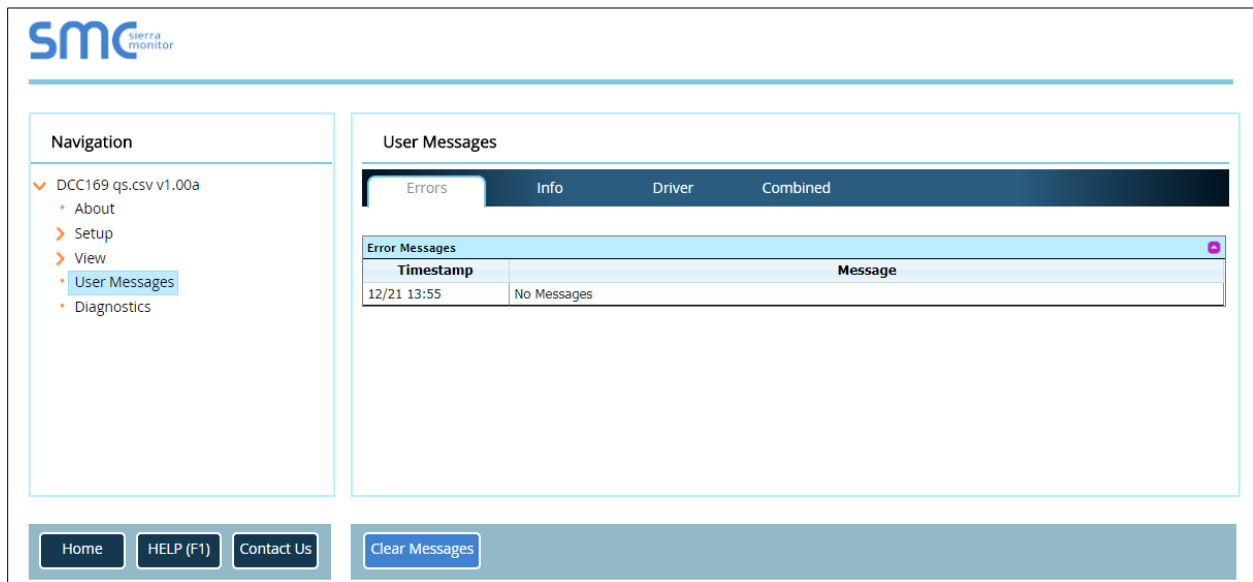
Error message #10 is very important. The driver prints this message when it suspects that the message contains important status information, but it doesn't know how to process it. If you record these messages and describe the events that produce them, FieldServer will be able to update the driver so that it can handle these messages correctly.

8.3.2   Catching Ignored Messages

Do not use this method if you intend viewing ignored messages in the way that is described in **Section 8.3.1**.

To assist you catch and monitor ignored messages the driver provides a special map descriptor keyword.

Make a map descriptor and set the parameter *AM6000_TypeID* 's value to *Catch-All*. Make sure that this is the last map descriptor in the csv file. The map descriptor requires a data length of at least 82 and when using FS-GUI you should view the data array in <S>tring mode.

The driver will place any ignored messages in this buffer. You will be able to read the message in <S>tring mode an make a decision on the necessary corrective action. If there are multiple messages being ignored the buffer will be overwritten.

You could use your PLC / control device / Scada to monitor the first byte of this data array and generate an alarm if the value is non-zero. Thus, even though a message has been ignored you system will know about it.

8.4 Error Messages

Message	Explanation
AM6000:#1 FYI. The MapDesc called <%s> is too short	Set the length of the MapDesc to 500 in the CSV file. Then download the updated CSV file to the FieldServer and reset the FieldServer for the changes to take effect.
AM6000:#2 FYI. You could have used a MapDesc called <%s> to expose diagnostic info.	This message if for your information only. It may safely be ignored. Read **Section 8.2** for more information on capturing driver stats.
AM6000:#3 Err. <%s> isnt valid in a csv file as a typeID	The keyword specified in the CSV file for the *AM6000_TypeID* parameter is invalid. **Section 5.3.2** has a list of valid keywords. Correct the CSV file. Then download the updated CSV file to the FieldServer and reset the FieldServer for the changes to take effect.
AM6000:#4 Err. Unknown TypeID <%s>	The keyword specified in the CSV file for the AM*6000_TypeID* parameter is invalid. **Section 5.3.2** has a list of valid keywords. Correct the CSV file. Then download the updated CSV file to the FieldServer and reset the FieldServer for the changes to take effect.
AM6000:#6 Err. Test file <%s> not found. AM6000:#7 Err. Diagnostic line ignored. %d	Report these errors to FieldServer technical support. Capture a log on the port connected to the Notifier panel and provide a copy of the configuration file you are using.
AM6000:#8 Err. Cant find MapDesc's on port %d	This message indicates that your configuration is invalid. A message was received from a Notifier panel on a port which doesn't have any Notifier MapDesc's defined. If for example, you have two panels – one per port, then you need to define two sets of MapDesc's to capture message from each panel. One set must connect to a node on the first port and the second set must connect to a node which is on the second port. Correct the configuration. Then download the updated CSV file to the Fieldserver and reset the FieldServer for the changes to take effect.
AM6000:#9 FYI. Msg below was ignored:	The driver prints a message that has been ignored. Messages can be ignored because the driver doesn't know how to process them or because no MapDesc has been found to use to store the information contained in the message. You can safely ignore this message. However, we recommend that you keep track of these ignored messages. If you think the driver should have processed the message then provide a snapshot, log and configuration file and report this message to FieldServer. **Section 8.3** has additional information.
AM6000:#10 Err. Msg below was ignored. Report to FieldServer.	See the notes for error #9 and read the notes in **Section 8.3**. Error 10 is printed when the driver thinks it has found a message that contains status information but doesn't understand the keyword which describes the status.
AM6000:#11 Err/FYI. Msg below was ignored. Report to FieldServer.	Error #11 is similar to Error 9. However, the driver thinks that this is possibly a message

8.5   Listing of System Trouble Messages

Msg Index	Message
0	!!! ILLEGAL!!!
1	MAINS TROUBLE
2	POWER LOW ON MAIN SUPPLY
3	POWER LOW ON AUXILIARY SUPPLY
4	POWER LOW ON BATTERY-CHARGER
5	POWER OVERVOLTAGE ON MAIN SUPPLY
6	POWER OVERVOLTAGE ON BATTERY-CHARGER
7	FAILURE OR OVERLOAD ON MAIN SUPPLY
8	FAILURE OR OVERLOAD ON AUXIL. SUPPLY
9	FAILURE ON BATTERY-CHARGER
10	DISCONNEC. BATT. OR FUSE FAILURE
11	BATTERY EXHAUSTED
12	BATTERY CHARGER UNBALANCED
13	AUX. SUPPLY DISCONNECTED
14	AUX. SUPPLY USER FUSE
15	MAIN SUPPLY USER FUSE
16	SIREN FUSE
17	EARTH TROUBLE
18	SIREN SUPERV.LINE INTERRUPTED
19	BATTERY FAILURE
20	WIRE CUT ON LINE
21	SHORT CIRCUIT ON LINE
22	PRINTER: OFF LINE/BUFFER FULL/PAPER END
23	COMMUNICATION ERROR ON
24	LOW VOLTAGE ON LINE
25	SIDE A OPEN ON LINE
26	SIDE B OPEN ON LINE
27	SYSTEM KEYPAD TROUBLE
28	INTERNAL PROGR.ENABLING JUMPER :ENABLED
29	SYSTEM START UP
30	CPU RESET OR WATCH-DOG FAILURE
31	CRT-TERMINAL : OFF-LINE
32	FLASH MEMORY ERROR ON
33	SIREN SUPERV.LINE
34	LINE
35	ANNUNC.