Chipkin
Automation Systems

Driver Manual

(Supplement to the FieldServer Instruction Manual)

FS-8705-13
TOA VS-900 Security Intercom Systems

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after May 1, 2001

© **Chipkin Automation Systems**, Box 211, 3495 Cambie St, Vancouver, BC, Canada, V5Z 4R3
▪ **Tel:** (866) 383-1657, ▪ **Fax:** (416) 915-4024 ▪
Email: dfs@chipkin.com▪ **Website:** www.chipkin.com

TABLE OF CONTENTS

1. TOA VS-900 Security Intercom Systems Serial Driver Description

The TOA VS-900 Security Intercom Systems Serial Driver allows the FieldServer to transfer data to and from devices over RS232 using TOA VS-900 Security Intercom Systems Serial Driver protocol.

The FieldServer can emulate a Client. As a client the driver can poll for status information and send commands to the FSU controller.

The driver is a serial driver using a RS232 serial port to connect between the FieldServer and the VS-900. An RS485 port together with a converter can also be used for the connection.

Server functionality is provided only to support our ongoing quality assurance program by facilitating automated testing of the driver. It is not documented or supported. If required please contact the FST sales group to discuss your requirements.

Max Nodes Supported

FieldServer Mode	Nodes	Comments
Client	1	Only 1 FSU-ESC node per connection
Server	0	Not supported or documented.

Supported Functions
Remote Dial
Log Data Request

2. Driver Scope of Supply

2.1. Supplied by FieldServer Technologies for this driver

FieldServer Technologies PART #	Description
-	No specific cables are shipped with this driver. A generic RJ45 Ethernet cable must be shipped with this driver.
-	A generic male and Female connector kit must be shipped with this driver.
FS-8705-13	Driver Manual.

2.2. Provided by the Supplier of 3rd Party Equipment

2.2.1. Required 3rd Party Hardware

Part #	Description

2.2.2. Required 3rd Party Software

2.2.3. Required 3rd Party Configuration

None Known.

3. Hardware Connections

Drawing to be furnished by Master Thermal Systems

3.1. Hardware Connection Tips / Hints

To be provided by Master Thermal Systems.

4. Configuring the FieldServer as a VS900 – Security Intercom System 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 FS).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a VS-900

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 FSC - Electronic Siren Controllers Serial 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.

Section Title
Data_Arrays

Column Title	Function	Legal Values
Data_Array_Name	Provide name for Data Array	Up to 15 alphanumeric characters
Data_Array_Format	Provide data format. Each Data Array can only take on one format.	Float, Bit, UInt16, Uint32, SInt16, Packed_Bit, Byte, Packed_Byte, Swapped_Byte
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,
DA_AI_01,                 UInt16,             200
DA_AO_01,                 UInt16,             200
DA_DI_01,                 Bit,                200
DA_DO_01,                 Bit,                200
```

4.2. Client Side Connection Descriptions

Create one connection for each VS900 port. Each connection can only be used to connect to a single VS900 interface/port.

Section Title
Connections

Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2[1]
Protocol	Specify protocol used	VS900
Baud*	Specify baud rate	Driver Supports : 110; 300; 600; 1200; 2400; 4800; **9600**; 19200; 28800; 38400; 57600; 115200 Baud Vendor Equipment Supports : 19200 Baud
Parity*	Specify parity	Driver Supports : Odd, Even, **None** Vendor Equipment Supports: **Even**
Data_Bits*	Specify data bits	Driver Supports : 7,**8** Vendor Equipment Supports : 8
Stop_Bits*	Specify stop bits	Driver Supports : 1,2 Vendor Equipment Supports: 1
Handshaking*	Specify hardware handshaking	**None**
Poll _Delay*	Time between internal polls	0-32000 seconds, **0.2 second**

[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.

VS900_Log_Timeout	When we poll for Data Logs to learn the status of any station we have to wait passively for the VS900 to sends them to us. We don't know how many there are so we after the final record we have to wait x milliseconds before we can send our next message. This is wasted time and results in lost data transmission capacity. This parameter controls how long we wait. It is worth optimizing. Specified in Milliseconds. If this parameter is not specified of if specified as zero then value of the Timeout parameter is used. Ensure this value is less than the Timeout parameter. The Timeout parameter is not normally specified. Its default is 2.0 seconds.	Default=0

Example

```
//        Client Side Connections

Connections
Port,          Protocol,           Baud,     Parity,      VS900_Log_Timeout,
P1,            VS900,              19200,    Even,        500,
```

4.3. Client Side Node Descriptors

Create one Node per connection only.

Section Title		
Nodes		
Column Title	**Function**	**Legal Values**
Node_Name	Provide name for node	Up to 32 alphanumeric characters
Node_ID	Station address of physical server node This parameter is not used directly by the driver. We recommend that a unique Node ID's be given to each node.	1-258
Protocol	Specify protocol used	VS900
Connection	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2[2]

Example

```
//    Client Side Nodes

Nodes
Node_Name            Node_ID            Protocol            Connection
SirenNode            1                  VS900               P1
```

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

[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.

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
Data_Type	Data type This commonly used parameter is not used by this driver.	
Length	Length of Map Descriptor This commonly used parameter is not used by this driver. However, it must be specified with a value of 1.	1
Address	This commonly used FieldServer parameter is not used by this protocol.	
VS900_Function	Specify the type of message that must be sent. If this parameters is omitted the driver assumes 'remoteDial'	**remoteDial** ReadLog
VS900_Exchange	The Number of the exchange that is being dialed or polled for log data. Valid positive integers are permitted.	1,2,3 …
VS900_Master	The Number of the Master Station doing the	1,2,3 ….

	dialing. This parameter is not required when polling for data logs. Valid positive integers are permitted.	
VS900_Remote	The number of the remote station being dialed. There is an alternate method of specifying the remote station. Read the notes in section xxxxx. The alternate method relies on this parameter not being specified or being specified as zero. This parameter is not required when polling for data logs. Valid positive integers and zero are permitted.	0,1,2,3 ….
VS900_CC	If this parameter is specified with a non-zero value then the driver will prefix each remote dial with a 'CC' which has the effect of cancelling any existing connection.	0,1
DA_Byte_Name	This parameter is used as part of the alternate method of specifying a dial string / remote station for a dial. Using this method it is possible to send any dial string. When this parameter is used the driver expects	One of the Data Array names from "Data Array" section above

	that the parameter VS900_Remote is set to zero or not specified. If the DA_Byte_Name parameter is specified then up to 32 characters of Dial String can be extracted from the specified Data Array and sent to the VS900. If this parameter is specified the parameter named VS900_Dial_String_Length must be specified too.	
VS900_Dial_String_Length	This parameter is used as part of the alternate method of specifying a dial string / remote station for a dial. Using this method it is possible to send any dial string. If the DA_Byte_Name parameter is specified then up to 32 characters of Dial String can be extracted from the specified Data Array and sent to the VS900. The driver uses the value of this length parameter to control the number of characters extracted from the Data Array. The driver pads the extracted characters with spaces to make a string 32 characters long.	

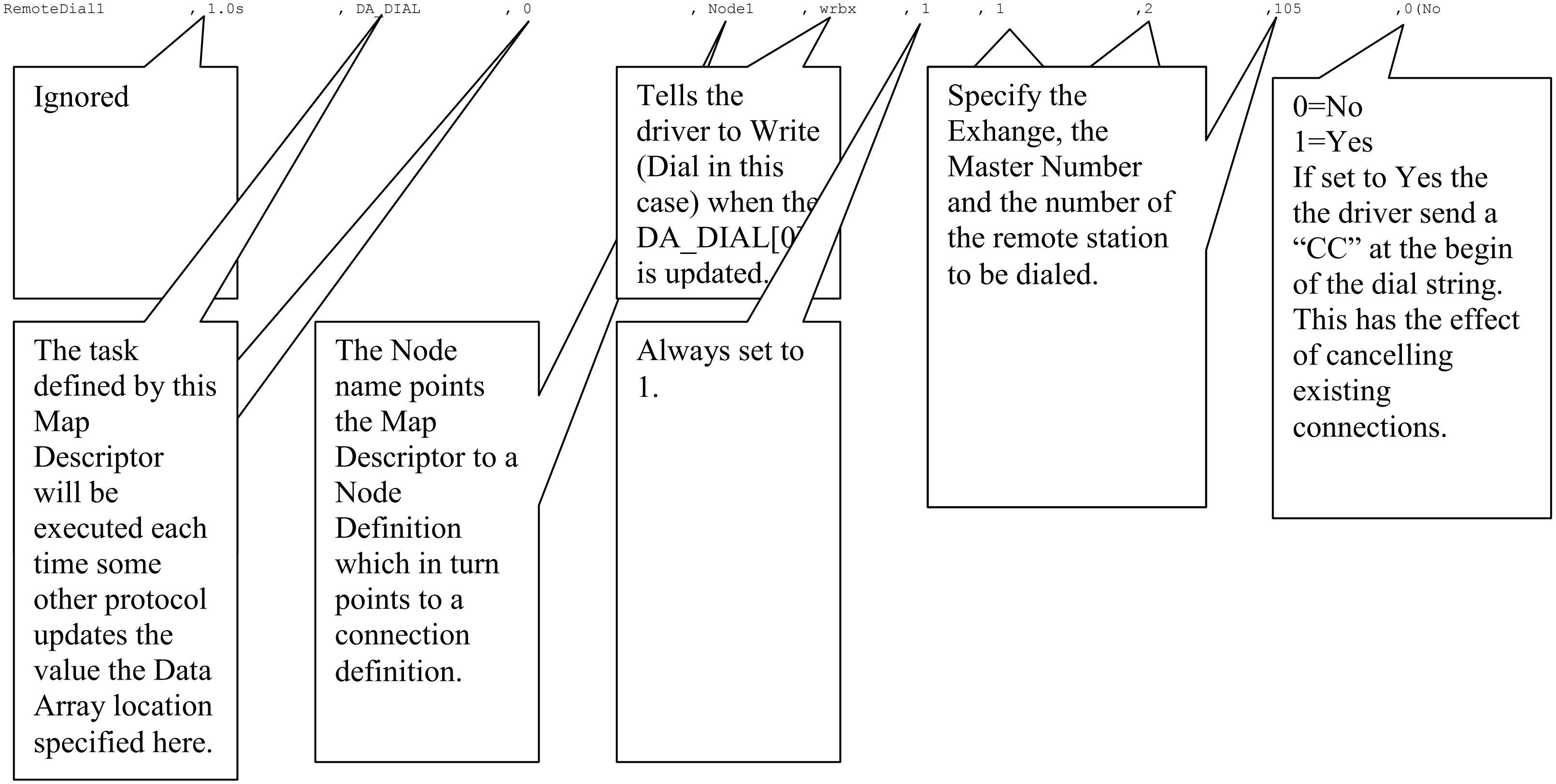
4.4.3. Timing Parameters

Column Title	Function	Legal Values
Scan_Interval	Rate at which data is polled	≥0.001s

4.4.4. Map Descriptor Example 1 – Dial.

In this example the driver send the command each time the Data Array location DA_DIAL[0] is updated. Typically that will be done by another protocol sending a command mapped to the same Data Array location. The driver will dial Exhange 1, Master1, Remote 105. A ‘CC’ string will not prefix the remote station.

```
Map_Descriptors
Map_Descriptor_Name , Scan_Interval , Data_Array_Name , Data_Array_Offset , Node_Name , Function , Length, VS900_Exchange ,VS900_Master ,VS900_Remote ,VS900_CC
RemoteDial1         , 1.0s          , DA_DIAL         , 0                 , Node1     , wrbx     , 1     , 1              ,2            ,105          ,0(No
```



Ignored

The task defined by this Map Descriptor will be executed each time some other protocol updates the value the Data Array location specified here.

The Node name points the Map Descriptor to a Node Definition which in turn points to a connection definition.

Tells the driver to Write (Dial in this case) when the DA_DIAL[0] is updated.

Always set to 1.

Specify the Exhange, the Master Number and the number of the remote station to be dialed.

0=No
1=Yes
If set to Yes the the driver send a "CC" at the begin of the dial string. This has the effect of cancelling existing connections.

Special Note for Remote Dial's : How to send a hang-up command

The data value found in the Data Array element associated with the Map Descriptor (Offset zero of the Data Array 'DA_Dial' in the above example) controls the message that will be sent.

If the value is 0 then the driver sends a 'CC' command to hang-up the connection.

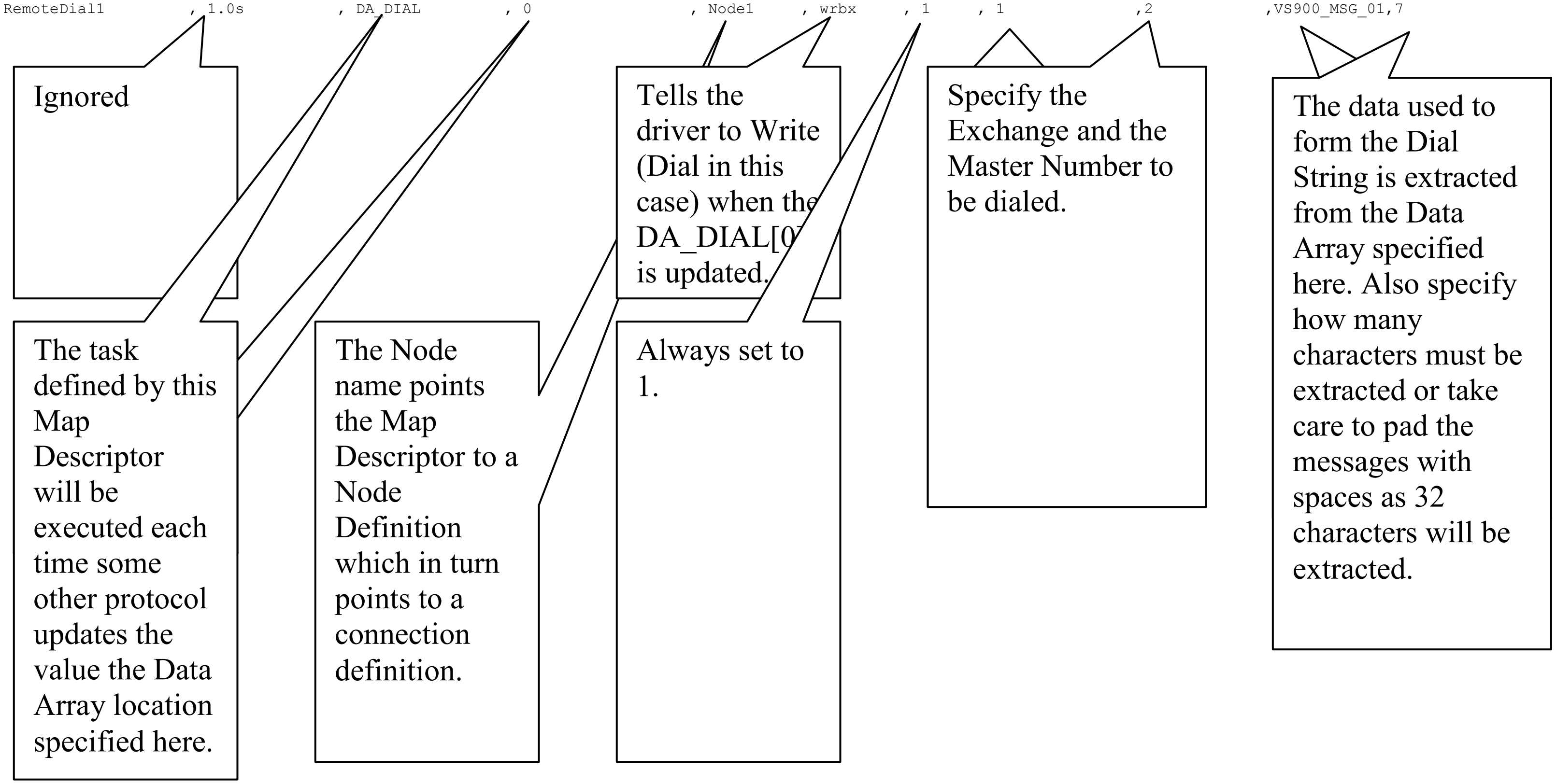
If the value is 99 then the driver extracts 32 bytes from a Data Array and sends those bytes. This allows you to specify the hag-up message. The driver extracts 32 bytes from the data array named 'vs900-stats' starting at offset 100. If the driver finds a zero in any one of these data array locations then the driver places a space in the message instead of a zero. To send the string "ABCD" you would load the value 65 into offset 100, 66 into offset 101… 68 into offset 103 and leave offsets 104-131 zero.

If the value is not zero or 99 then the driver sends the dial command as illustrated in examples 1 and 2.

4.4.5. Map Descriptor Example 2 – Dial with User controlled Dial String

In this example a 2ndary Data Array is specified. The dial string is formed by extracting data from the 2ndary Array. A maximum of 32 characters can be send. There is a way to preload data. See Example 3. The Data Array can be populated with data preloaded from the configuration file, using a FieldServer script (Additional Support may be required.) or by having a device using another protocol write to the Data Arrays. This latter method is volatile. Data will be lost on restart.

```
Map_Descriptors
Map_Descriptor_Name , Scan_Interval , Data_Array_Name , Data_Array_Offset , Node_Name , Function , Length, VS900_Exchange ,VS900_Master ,DA_Byte_Name, VS900_Dial_String_Length
RemoteDial1         , 1.0s          , DA_DIAL         , 0                 , Node1     , wrbx     , 1     , 1              ,2            ,VS900_MSG_01,7
```



Ignored

The task defined by this Map Descriptor will be executed each time some other protocol updates the value the Data Array location specified here.

The Node name points the Map Descriptor to a Node Definition which in turn points to a connection definition.

Tells the driver to Write (Dial in this case) when the DA_DIAL[0] is updated.

Always set to 1.

Specify the Exchange and the Master Number to be dialed.

The data used to form the Dial String is extracted from the Data Array specified here. Also specify how many characters must be extracted or take care to pad the messages with spaces as 32 characters will be extracted.

4.4.6. Example 3 – Pre-loading a Data Array

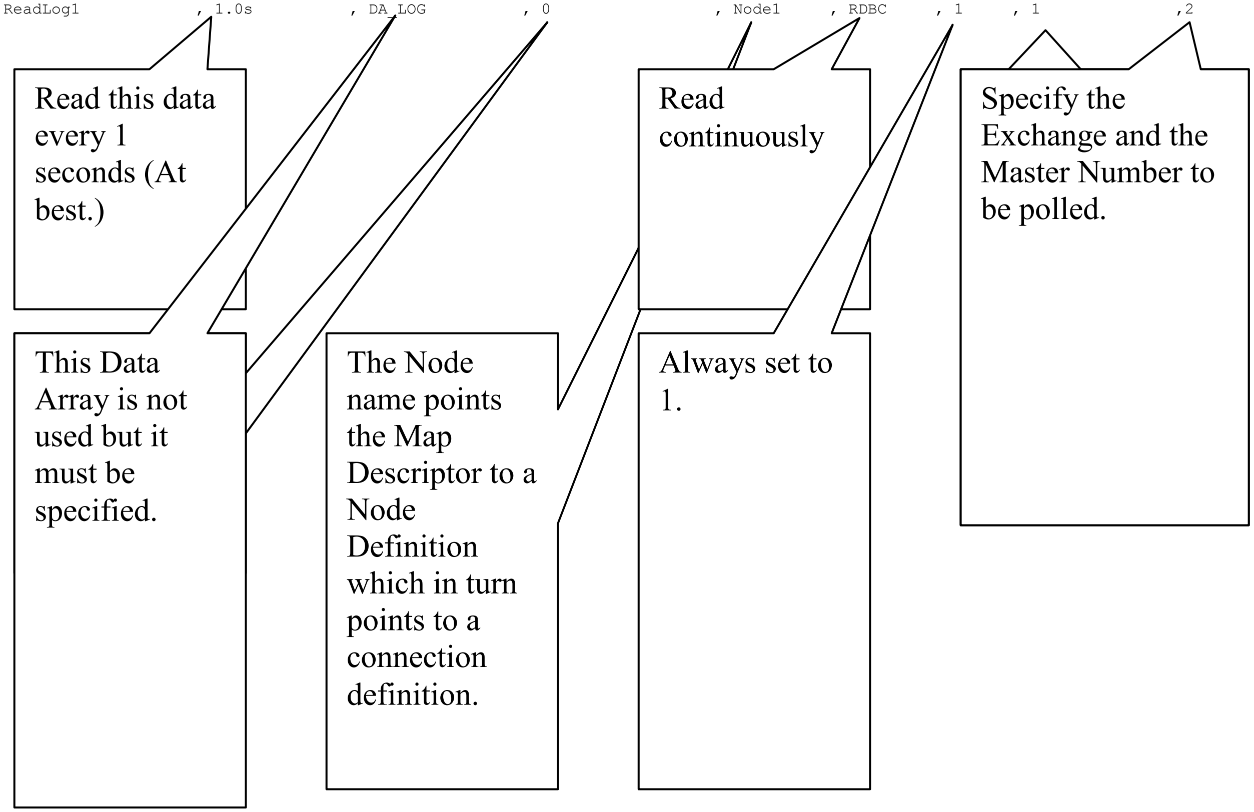
In the example below the 1st seven elements of the Data Array are loaded with the indicated values. Specifically the values 67,67,42,56,48,48,35 are the decimal equivalents of the ASCII Characters **CC*800#** If this data was preloaded and used in conjunction with the Map Descriptor 2 example then an All-Call Paging would be initiated after terminating any current Master Station functions.

```
Preloads,
Data_Array_Name ,Preload_Data_Index ,Preload_Data_Value
VS900_MSG_01    ,00                 ,67
VS900_MSG_01    ,01                 ,67
VS900_MSG_01    ,02                 ,42
VS900_MSG_01    ,03                 ,56
VS900_MSG_01    ,04                 ,48
VS900_MSG_01    ,05                 ,48
VS900_MSG_01    ,06                 ,35
```

4.4.1. Map Descriptor Example 4 – Poll for Data Logs – Getting Station Status

In this example Master 2 on Exchange 1 is polled for its data logs. This is done once per second. If data logs are sent in response to the poll then the data is stored in a Data Array whose name is derived from the Exchange and Station number. More information is provided in section 4.5

```
Map_Descriptors
Map_Descriptor_Name , Scan_Interval , Data_Array_Name , Data_Array_Offset , Node_Name , Function , Length, VS900_Exchange ,VS900_Master  ,VS900_Function
ReadLog1            , 1.0s          , DA_LOG          , 0                 , Node1     , RDBC     , 1     , 1              ,2             ,readLog
```



Read this data every 1 seconds (At best.)

This Data Array is not used but it must be specified.

The Node name points the Map Descriptor to a Node Definition which in turn points to a connection definition.

Read continuously

Always set to 1.

Specify the Exchange and the Master Number to be polled.

4.5. How Data Logs are Stored.

Data Logs are used to report the status of a particular station. When an exchange is polled for data logs the exchange can respond with zero or more data log reports. It depends how many events have occurred that involved that exchange. Don't confuse an operator action with an event. An event is a discrete action performed by the VS900. Only the vendor can report to you the correlation between actions and the events that form Data Log. For example, placing a normal call from a master to a remote station involves 5 events – Example. Station 200 calls Station 105. Call from Normal Sub Station, Station Call Reception, Call Operation Completion, Start of Normal Conversation, Normal Call Termination. Each one of these Data Logs involve 2 stations (in this example but up to 4 stations may be involved.). We will store the Status and the time twice for each of these Data Log events. One for station 200 and one for 105. If all 5 events had completed since we last polled the exchange then we will get all 5 data log messages in a row and thus the status of the last one (the most recent) will overwrite the data stored for the previous events. Thus, if the status of each station was monitored then a status reporting Normal Call Termination would be reported.

Each Data log reports the following ;

Exchange
Time
Status
Up to 4 Station Numbers. These are interpreted according to the table provided in the vendor Manual a copy of which is included in this manual.

The driver stores up to 4 data sets per Data Log – one for each of the 4 possible station numbers.

Using the Exchange number and the Station Numbers the driver builds a Data Array name.

Name = "VS9_EXxx_STNyyy"
Where xx = the exchange number. "00"-"99"
yy = the station number. Example "009"

Example: For Station 105 on Exchange 1 the Data Array Name is "VS9_EX01_STN105"

The driver looks for this array and stores the following table of data:

Table 1: How Status Report Data is stored.

Offset	Meaning	Notes
0	Reported Status Code	Section 4.6 provides more information. Remote devices should monitor this location in the Data Array to know the current status of each station on each exchange.
1	Interpreted Status Code	Section 4.6 provides more information.
2	Time Stamp - Hours	
3	Time Stamp - Minutes	
4	Time Stamp – Seconds	
5	Line 1 Number	
6	Line 2 Number	
7	Line 3 Number	
8	Line 4 Number	
10	Record x of 4	

4.6. Status Codes – Re-Map using a Lookup Table.

If a file called vs900.ini is present on the FieldServer it will be loaded to form a lookup table to map reported status's to an interested value.

The format of the file is as follows: The first column contains the reported status and the 2nd column contains the interpreted status. If the VS900 reports a status of 2 then the driver set the interpreted status to 33 (in this example.)

```
#lines that begin with a '#' are ignored
01,01,
02,33,
03,03,
04,04,
05,05,
06,06,
```

4.7. VS900 Status Codes.

State Code	Description	1of4 Station #	2of4 Station #	3of4 Station #	4of4 Station #
01	Call from Normal sub-station.	Calling Stn #	Called Stn #		
02	Call from Emergency sub-station.	Calling Stn #	Called Stn #		
03	Call from Master Station (Telephone Master)	Calling Stn #	Called Stn #		
04	Call from C/O line "Axxy " ("xx" is exchange #, and "y" is C/O line # 1 - 2	C/o line #	Called Stn #		
05	Master Station (Telephone Master) reception of a call from Normal sub-station	Master Stn #	1st Waiting Stn #		
06	Master Station (Telephone Master) reception of a call from Emergency sub-station	Master Stn #	1st Waiting Stn #		
07	Master Station (Telephone Master) reception of a call from other Master Station (Telephone Master)	Master Stn #	1st Waiting Stn #		
08	Master Station (Telephone Master) reception of a C/O line call	Master Stn #	1st Waiting Stn #		
09	Call operation completion at Master Station (Telephone Master)	Master Stn #	1st Waiting Stn #		
10	Start of Normal conversation	Calling Stn #	Called Stn #	Hold Stn #	
11	Start of Emergency conversation	Calling Stn #	Called Stn #	Hold Stn #	
12	Start of C/O line conversation	Calling Stn #	Called Stn #	Hold Stn #	
13	Automatic response to an incoming C/O line call (direct-in dial) "Axxy " ("xx" is exchange #, and "y" is C/O line # 1 - 2	C/o line #			
14	Start of Individual-zone or All-zone Paging "pagxx " ("xx" is paging zone #01 - 19)	Calling Stn #	Paging #		
15	Start of Emergency All-zone Paging "pagxx " ("xx" is paging zone #01 - 19)	Calling Stn #	Paging #		
16	Start of External Broadcast activation	Activated Input #	Paging #		

	"chx " ("x" is the external line # 1 - 4)				
17	Connection of Scan Monitor	Operating Stn #	Monitored Stn #		
18	Not used				
19	Start of Conference call	Call originating Stn #			
20	Connection of Conference call	Call originating Stn #	Participant Stn #	Participant Stn #	
21	Start of Emergency Conference call	Call originating Stn #			
22	Connection of Emergency Conference call	Call originating Stn #	Participant Stn #	Participant Stn #	Participant Stn #
23	Automatic Call Forward log	Stn # to xfer a call	Stn # all is xferred to		
24 - 29	Not Used				
30	Call termination by disappearance of waiting stations. Displayed when reception mode is switched to standby mode.	Master Stn #			
31	Call termination before dialing completion	Operating Stn #	Called Stn #		
32	Normal call termination	Calling Stn #	Called Stn #	Hold Stn #	
33	Receiving C/O line call interruption “Axxy " ("xx" is exchange #, and "y" is C/O line # 1 - 2	C/o line #			
34	Paging termination	Calling Stn #	Paging #		
35	Emergency Paging termination	Calling Stn #	Paging #		
36	Termination of external input broadcast. "chx " ("x" is the external line # 1 - 4)	Calling Stn # Activated Input #	Paging #		
37	Termination of Scan Monitor	Operating Stn #	Monitored Stn #		
38	Termination of Conference	Call originating Stn #	Participant Stn #	Participant Stn #	
39	Termination of Emergency Conference	Call originating Stn #	Participant Stn #	Participant Stn #	Participant Stn #
40	Line disconnection. Displayed when only a single line for call transfer or call-back is disconnected.	Disconnected Stn #			
41	Call termination due to the ringing repetition limit	Calling Stn #	Called Party Stn #		

	(No-Answer)				
42	Call termination due to the time limit (conversation, C/O and Paging)	Calling Stn #	Called Party Stn #	Hold Stn #	
43	Call termination (at the called station). Displayed when a received call disappears.	Calling Stn #			
44	Forced call termination (Priority)	Calling Stn #	Called Party Stn #	Hold Stn #	

5. Configuring the FieldServer as a VS900 – Security Intercom System Client

This driver has a server side implemeted but it is used for FieldServer's Quality Assurance program and is not docuemnted or supported. If you are interested in using Sever Side features then please contact Chipkin Automation Systems.

Appendix 1. Advanced Topics

This section is blank.

Appendix 2. Troubleshooting tips

2.1.1. Connection Tips & Hints

This section is blank.

2.1.2. Driver Error Messages

Error Message We have shown place holders for the parts of the message which change. %s is a place holder for a text string. %d is a place holder for a number %c is a place holder for an alpha character.	**Explanation and corrective action**
VS900:#01 Err. Cant find DA=%s	Take a log using FST_Diag and then call Tech Support.
VS900:#02 Err. Cant open rqd file=%s	The file vs900.ini is required by the driver. Read section 4.6
VS900:#03 Err. Cant find DA=%s	The driver has received a data Log Message and cant find an appropriate Data Array to store the data. Read section on how to form appropriate Data Array Names. Add a Data Array who name is reported in this message to the config. Download the modified config and reset the FieldServer for the changes to take effect.
VS900:#04 Err. Error#%d Parsing Log Message. Dump Follows	If this error is printed often then take a log using FST_Diag and then call Tech Support. If not, you could assume it to be the product of an occasional corrupted message.
VS900:#05 Err. Error#%d Parsing Log Message. Dump Follows	If this error is printed often then take a log using FST_Diag and then call Tech Support. If not, you could assume it to be the product of an occasional corrupted message.
VS900:#06 Err. Max number of states=%d	The maximum number of states that can be loaded from vs900.ini is 100. Read section 4.6
VS900:#07 Err. Remote=0 and DA_Byte_Name=". Cant send.	Read section 4.4.2. If the VS900_Remote parameter is specified with a value of zero then the

	driver expects the parameter called DA_Byte_Name to be specified.
VS900:#08 FYI. Use an Array called <%s> to expose diagnostic info.	Read section 2.1.3
VS900#10 Store:%s (%d)	Printed each time a data log message is stored. The message and the reported state are printed.
VS900:#20 FYI We suggest use of connection param=VS900_Log_Timeout.	Read Section 4.4.2
TLC:#46 Err. Timeout waiting for ENQ.	If this error is printed often then take a log using FST_Diag and then call Tech Support. If not, you could assume it to be the product of an occasional corrupted message.

2.1.3. Exposing Driver Stats

The diver makes some of its operating statistics available in a Data Array where they can be read by an upstream device. The lines from the example below can be cut and pasted into a configuration file.

```
Data_Arrays,
Data_Array_Name,        Data_Format,          Data_Array_length,
VS900-stats,            UINT32,               1000,
```

Offset	Description
1	Incremented each time the client sends an Ack
2	Incremented each time the client sends any Message
3	Total number of bytes the client has sent
4	Incremented each time the client sends an EOT
5	Incremented each time the client sends a Remote Dial Msg
6	Incremented each time the client sends an Data Log Request
7	Incremented each time the client receives an Ack
8	Total number of bytes the client has received from the VS900
9	Incremented each time the client sends an Enq
10	Incremented each time the client receives an Enq
11	Incremented each time the client receives an EOT (completes Data Log Record session)
12	For test purposes only.
13	For test purposes only.

Appendix 3. Revision History

Date	Resp	Format	Driver Ver.	Doc. Rev.	Comment
24 Dec 2008	PMC		1.00a	0	Document Created
14 Feb 2009	PMC		1.01a	1	Updated for new method to send hang-up Updated with new MD parameter named VS900_Function.
14 July 2009	PMC		1.01a	2	Report and Interpreted. Reported at offset zero, Interpreted at offset 1. Previously was swapped.