



FieldServer
FS-8700-49 Data Aire (DAP)
Driver Manual
(Supplement to the FieldServer Instruction Manual)

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after November 2015

Kernel Version: 1.07
Document Revision: 9

Contact Information:

Thank you for purchasing the FieldServer.

Please call us for Technical support of the FieldServer product.

Contact Information:

Sierra Monitor Corporation
1991 Tarob Court
Milpitas, CA 95035

Contact number:
+1 408 262-6611
+1 800 727-4377

Email: info@sierramonitor.com

Website: www.sierramonitor.com

TABLE OF CONTENTS

- 1 Data Aire Description 5**
 - 1.1 Performance Issues – DARTIII only 5
 - 1.2 Data Alarm Network Module Network LED Operation 5
- 2 Driver Scope of Supply 5**
 - 2.1 Supplied by Sierra Monitor Corporation for this driver 5
- 3 Hardware Connections..... 6**
- 4 Configuring the FieldServer as a Data aire Client 7**
 - 4.1 Data Arrays/Descriptors 7
 - 4.2 Client Side Connection Parameters 8
 - 4.3 Client Side Node Parameters 9
 - 4.4 Client Side Map Descriptor Parameters 10
 - 4.4.1 *FieldServer Specific Map Descriptor Parameters* 10
 - 4.4.2 *Driver Related Map Descriptor Parameters* 10
 - 4.4.3 *Timing Parameters* 10
 - 4.4.4 *Map Descriptor Example.* 11
 - 4.4.5 *Map Descriptor: Example 2 – Writing a Set-Point.* 11
 - 4.4.6 *Map Descriptor: Example 3 – Writing multiple points using one message.* 12
- Appendix A. Data Tables 13**
 - Appendix A.1. Data Table 1: Array Locations of ‘Everything’ 13
 - Appendix A.2. Data Table 2 - Point Descriptions 18
 - Appendix A.3. Data Table 3 - Unit Types 24
- Appendix B. Advanced Topics 25**
 - Appendix B.1. Additional Driver Specific Map Descriptor Parameters..... 25
 - Appendix B.2. DA_Func Parameter - Permitted values. 26
 - Appendix B.3. DA_Field Parameter - Permitted values. 27
 - Appendix B.4. DA_Method Parameter Values and Notes..... 28
 - Appendix B.5. Advanced Map Descriptor 1:Collecting Data From Multiple Fileds 29
 - Appendix B.6. Advanced Map Descriptor Example 2:Collecting Error Messages 30
 - Appendix B.7. Advanced Map Descriptor Example 3: Using Special DA Field 31
 - Appendix B.8. Advanced Map Descriptor Example 4: - Using DA_Assoc 32
 - Appendix B.9. Advanced Map Descriptor: Example 5 - Using a special / diagnostic command.: 33
 - Appendix B.10. Advanced Map Descriptor: Example 6 – Turning Dap’s On/Off. 33
- Appendix C. Troubleshooting Tips 34**
 - Appendix C.1. Bad Values 34
 - Appendix C.2. Dead Nodes 34
 - Appendix C.3. Noise 34
 - Appendix C.4. Ignored Messages 34
 - Appendix C.5. Driver limitation 34
 - Appendix C.6. Baud Rate..... 34
 - Appendix C.7. Wiring..... 35
 - Appendix C.8. Stuck Unit. 35
 - Appendix C.9. Number of Units 35
- Appendix D. Error Messages 36**
 - Appendix D.1. Exposing Driver Statistics 39
- Appendix E. Writing data to DAP Devices 40**
 - Appendix E.1. Background..... 40
 - Appendix E.2. Associated Map Descriptors 40
 - Appendix E.3. Starting/Stopping DAP devices..... 41

Appendix E.4. DAP Unit Errors 42
Appendix E.5. Using DA_Assoc correctly with Writes/Toggling Values..... 47
Appendix F. Write Through's 48

LIST OF FIGURES

Figure 1 - Generic Connection Diagram 6

1 DATA AIRE DESCRIPTION

The Data Aire Driver is capable of direct connection to a network of Data Aire DAP devices. The FieldServer is connected in a RS-485 loop topology. Two FieldServer ports are required per loop. The driver is an active client. All data is obtained by poll & response.

Connection to a DART is not supported by this driver.

The driver may be configured very simply (See Section 4.4). A number of advanced configurations are also available and are described in Appendix B. The driver supports the common message formats for common Data Aire DAP devices. A list of the supported messages is provided in the manual.

The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer.

1.1 Performance Issues – DARTIII only

Data Aire communication is based on a very low baud rate. In addition inter-message timing constraints and overhead requirements for active messages in a Dart configuration add significant time to each transaction. It is not possible to write a setpoint to a device until the device has been successfully read, and the result of the write will not be seen until the next read is complete. The result of a write may not be seen for some time due to slow communications and significant inter poll delays.

1.2 Data Alarm Network Module Network LED Operation

DAP-II units communicate with a DART unit by using a Data Alarm Network Module. This module provides a RS-485 communications link. Installed on the card are several LEDS. The table below explains how the LEDS appear under normal (correctly functioning) operation.

LED	LED Appearance
FUSE	The Network Module has a blown fuse detection circuit. If one (or more) of the fuses has an open circuit, then the FUSE LED will be lit. Normally, this LED is dark.
DS1 & DS3	Communications signals enter and leave the Network Module. These LEDS reflect the communications activity. When first powered up, both LED's will be lit and the illumination will be steady. As the DART communicates with the network, both LED's will begin to flicker in unison. All of the Network Module LED's will flicker at the same time.
NETWORK	At some point the DART will send a communication message to a specific network-connected DAP-II that requires a reply to the DART. If the message is received without error, the DAP-II will reply by engaging a relay on the Network Module and transmitting a message. When this happens the NETWORK LED will come on. Whenever a NETWORK LED is seen to come on it means that the communications into the Network Module is satisfactory. It is still possible to have a wiring problem downstream of the module that prevents the message from being received by the DART

2 DRIVER SCOPE OF SUPPLY

2.1 Supplied by Sierra Monitor Corporation for this driver

Sierra Monitor Corporation PART #	Description
	RS-485 connection adapter

3 HARDWARE CONNECTIONS

The FieldServer is connected to the DAP devices as shown in connection drawing.

Configure the DAP devices according to manufacturer's instructions.

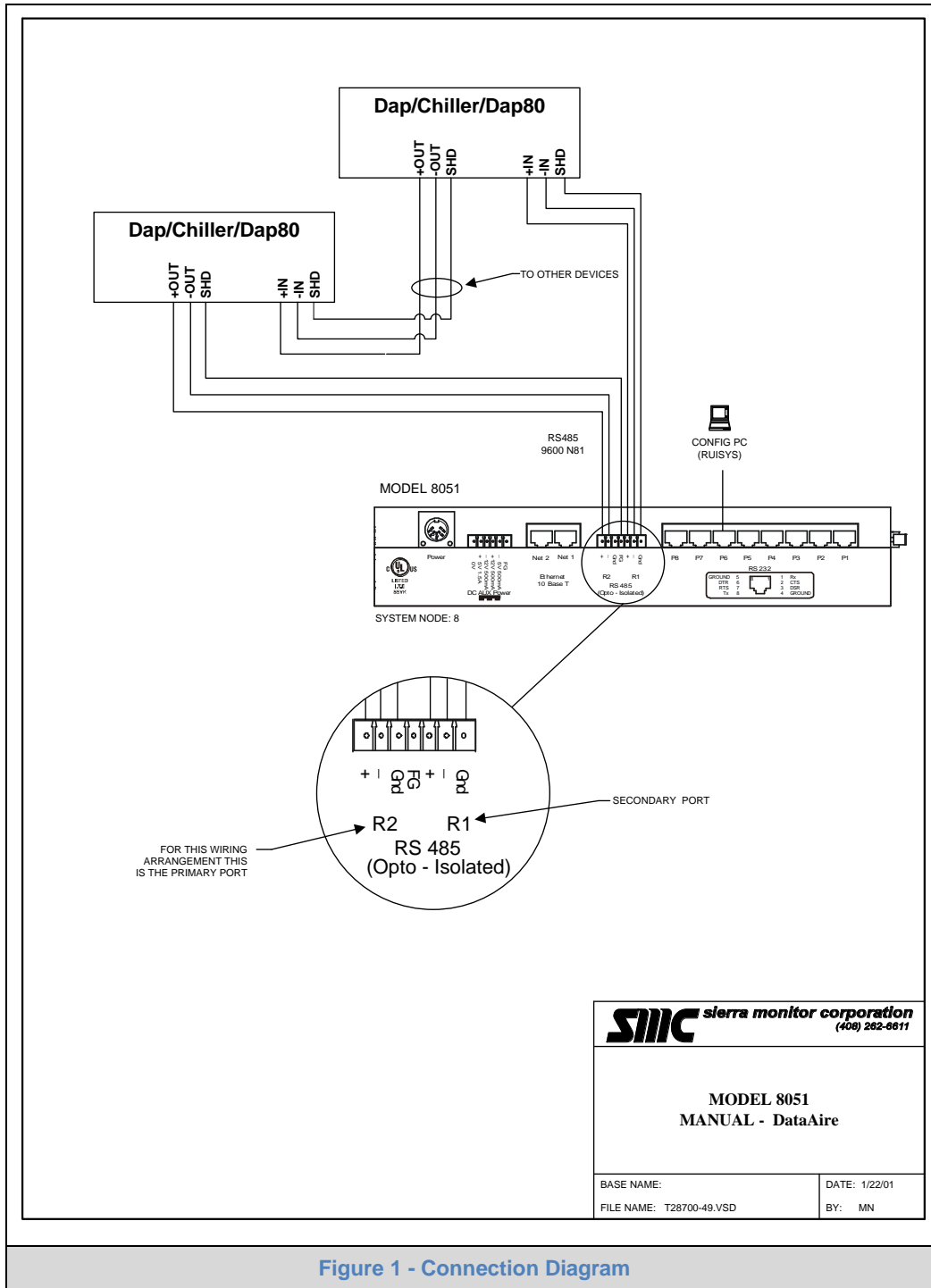


Figure 1 - Connection Diagram

4 CONFIGURING THE FIELDSEVER AS A DATA AIRE CLIENT

For a detailed discussion on FieldServer configuration, please refer to the FieldServer instruction 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 Data Aire Server.

4.1 Data Arrays/Descriptors

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, 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 Parameters

Section Title		
Connections		
Column Title	Function	Legal Values
Port	This is the FieldServer port connected to the DART's RS-232 port.	P1-P8, R1-R21 (P1-P8) requires 232/485 converter.
Secondary_Port*	The return loop from the DAP devices is connected to this port. (Loop connected to Tx+ & Tx-terminals.	P1-P8, R1-R21 (R1-R2) requires 232/485 converter.
Baud*	Specify baud rate	2400/96002 (Vendor Limitation)
Parity*	Specify parity	None
Data_Bits*	Specify data bits	8
Stop_Bits*	Specify stop bits	1
Protocol	Specify protocol used. Either keyword may be used.	DAP, Daire
Handshaking*	Specify hardware handshaking	None
Poll Delay*	Time between internal polls	0-32000s, 1s

Example

```
// Client Side Connections

Connections
Port , Secondary Port , Protocol , Baud , Parity , Data_Bits , Stop_Bits , Poll_Delay
R1 , R2 , DAP , 2400 , None , 8 , 1 , 0.100s
```

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.

² DAP V9.5 or later supports 9600

4.3 Client Side Node Parameters

Section Title		
Nodes		
Column Title	Function	Legal Values
Node_Name	Provide name for node	Up to 32 alphanumeric characters
Node_ID	Modbus station address of physical server node	1-259
Protocol	Specify protocol used	DAP
Connection	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2 ³

Example

```
// Client Side Nodes
Nodes
Node_Name , Node_ID , Protocol , Connection
Unit1 , 1 , DAP , R1
```

This is the connection to the primary port identified with the 'Port' keyword in the connection configuration

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.

4.4 Client Side Map Descriptor Parameters

4.4.1 FieldServer Specific Map Descriptor Parameters

Column Title	Function	Legal Values
Map_Descriptor_Name	Name of this Map Descriptor	Up to 32 alphanumeric characters
Data_Array_Name	Name of Data Array where data is to be stored in the FieldServer	One of the Data Array names from "Data Array" section above
Data_Array_Offset	Starting location in Data Array	0 to (Data_Array_Length-1) as specified in "Data Array" section above
Function	Function of Client Map Descriptor	Rdbc, Wrbc, Wrbx

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	Register, Coil, AI, DI
Length	Length of Map Descriptor	1
Address	Starting address of read block	Not required for client configuration - used in simulation configuration only.
DA_Func*	Only required when the function is a Read (rdbc). Tells the driver to poll the device for all the data that is available. The data is device specific. Other uses of this parameter are discussed in Appendix B .	Everything – See Appendix A
DA_Field	Required when the function is a write (wrbc) or when the Map Descriptor is associated with a write by means of the DA_Assoc parameter value. This is the name of the data field whose value you wish to set in the device. Other uses of this parameter are discussed in Appendix B.	See Appendix A Appendix A.2 for a list of permitted values.
DA_Assoc	Use to associate passive Map Descriptors with an active Map Descriptor. Multiple fields can be associated with one WRBC Map Descriptor, thus writing multiple values to one device using just one message.	Any positive integer. See Appendix E

4.4.3 Timing Parameters

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

4.4.4 Map Descriptor Example.

The FieldServer listens passively for all possible data from the device called 'unit1' and stores the data in an array called 'DA_01'. The data that is obtained from 'unit1' is dependent on the type of device. Irrespective of the device type the arrangement of data, stored in DA_01, is fixed. If a data field cannot be obtained from 'unit1' then the array is left with a zero value for that data field.

```
// Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Address , Length , DA_Func , Scan_Interval
Read_Node_01 , DA_01 , 0 , RDBC , Unit1 , 0 , 1300 , Everything , 2.5s
```

It is recommended that the format of the Data Array be FLOAT as the data read from the unit consists of bits, bytes and real numbers.

The node name connects the Map Descriptor to a node definition which in turn is connected to a port.

Always set the Address to zero and the length to 1300 as this is the max number of data points that can be read

Tells the driver that this Map Descriptor is used to (poll &) store a complete data set from the unit.

The scan interval is used to control the minimum time between poll messages. It may take several polls / messages to obtain a complete data set (the number is dependent on the unit type.)

4.4.5 Map Descriptor: Example 2 – Writing a Set-Point.

```
// Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Address , Length , DA_Field , Scan_Interval
Write_SP_01 , DA_SETPOINTS , 0 , WRBC , Unit1 , 0 , 1 , nom_h_setpt , 120s
```

The setpoint is obtained from this Data Array.

Always leave the address set to zero and the length as one.

Select the data field from Appendix B.3. Ensure that the keyword can be written to the unit type identified as unit1. Example: If unit1 is a Chiller then the above Map Descriptor will not function correctly as it can only be used to write to unit types 7, 9, 14, 15.

This is how often the data will be written to the device. Reduce the scan interval when using a dart. The dart will only take control of the devices when the network has been idle for at least 50 seconds. If polls are generated too often, the dart will never exercise control.

4.4.6 Map Descriptor: Example 3 – Writing multiple points using one message.

Additional information is provided in Appendix B

```
// Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name    , Data_Array_Name    , Data_Array_Offset    , Function    , Node_name    , Address    , Length    , DA_Field    , DA_Assoc    , Scan_Interval
Write_Map Descriptor_1 , DA_SETPOINTS    , 0                    , WRBC        , Unit1        , 0          , 1         , nom_h_setpt , 2           , 120s
Write_Map Descriptor_1 , DA_SETPOINTS    , 1                    , passive     , Unit1        , 0          , 1         , t_setpt     , 2           ,
```

A 'Write' and a passive to the same node. This write will update two fields, the nom_h_setpt and the t_setpt. The write Map Descriptor. must precede the passive. For a Dart, use wrbx instead of wrbc.

Associate the passive Map Descriptor to the active (wrbc) Map Descriptor. In this way the driver will use only one message to write to the device. The message will be built using both Map Descriptors. This method reduces the communication load.

The association is made using the DA_Assoc parameter. Use unique positive integers. Omitting the DA_Assoc parameter when using 'passive' Map Descriptors will produce ambiguous results.

Appendix A. DATA TABLES

Appendix A.1. Data Table 1: Array Locations of 'Everything'

In the following table the array location indicates the offset in the Data Array at which a data field can be found. (This offset is relative to the offset specified in the Map Descriptor) The columns headed 2, 3 ... indicate the unit types for which the data fields are available. For example: The field 'd_temp' can be read from unit types 2, 5,6,7,9 but not from any of the other unit types. It is beyond the scope of this manual to describe each field and to indicate valid ranges. Such information should be obtained from the Data Aire Corporation.

'x' Indicates Read only

'X' Indicates a point that can be read & written.

'w' Indicates a write only point.

Array	Method	Num	Data Field	2	3	4	5	6	7	8	9	14	15	Message Type
1	1	1	Zone		w	w	w	w	w	w	w	w	w	Dap-Config
2	1	1	Inhibit		w	w	w	w	w	w	w	w	w	Dap-Config
3	4	1	unitType	x	x	x	x	x	x	x	x	x	x	Dap-Unit
4	2	1	temp	x			x	x	x		x			Dap-Stat
5	2	1	hum	x			x	x	x		x			Dap-Stat
6	2	1	d_temp	x			x	x	x		x			Dap-Stat
7	3	8	mode	x			x	x	x		x			Dap-Stat
23	3	8	hold	x			x	x	x		x			Dap-Stat
39	1	1	cs_on	x			x	x	x		x			Dap-Stat
40	1	1	hs_on	x			x	x	x		x			Dap-Stat
41	1	1	valvePCT	x			x	x	x		x			Dap-Stat
42	1	1	hVlvPCT	x			x	x	x		x			Dap-Stat
43	4	14	Errors - see Appendix E.4 for more information	x			x	x	x		x			Dap-Stat
57	2	1	hiTemp	x			x	x	x		x			Dap-Stat
58	2	1	loTemp	x			x	x	x		x			Dap-Stat
59	2	1	hiHum	x			x	x	x		x			Dap-Stat
60	2	1	loHum	x			x	x	x		x			Dap-Stat
61	1	1	chilled_water	x			x	x	x		x			Dap-Stat
62	1	1	compressor_config	x			x	x	x		x			Dap-Stat
63	1	1	heat_strip_config	x			x	x	x		x			Dap-Stat
64	1	1	hum_config	x			x	x	x		x			Dap-Stat
65	1	1	csUtilPct	x			x	x	x		x			Dap-Stat
66	1	1	hsUtilPct	x			x	x	x		x			Dap-Stat
67	1	1	valveUtilPct	x			x	x	x		x			Dap-Stat
68	1	1	humUtilPCT	x			x	x	x		x			Dap-Stat
69	1	1	alrm_select_1	x			x	x	x		x			Dap-Stat

Array	Method	Num	Data Field	2	3	4	5	6	7	8	9	14	15	Message Type
70	1	1	alarm_select_2	x		x	x	x	x	x				Dap-Stat
71	1	1	alarm_select_3	x		x	x	x	x	x				Dap-Stat
72	1	1	alarm_select_4	x		x	x	x	x	x				Dap-Stat
73	3	56	bitErrors - see Appendix E.4 for more information	x		x	x	x	x	x				Dap-Stat
185	1	1	runtime_c1	x		x	x	x	x	x				Dap-Xtra
186	1	1	runtime_c2	x		x	x	x	x	x				Dap-Xtra
187	1	1	runtime_c3	x		x	x	x	x	x				Dap-Xtra
188	1	1	runtime_c4	x		x	x	x	x	x				Dap-Xtra
189	1	1	runtime_ht1	x		x	x	x	x	x				Dap-Xtra
190	1	1	runtime_ht2	x		x	x	x	x	x				Dap-Xtra
191	1	1	runtime_ht3	x		x	x	x	x	x				Dap-Xtra
192	1	1	runtime_hum	x		x	x	x	x	x				Dap-Xtra
193	1	1	runtime_evap	x		x	x	x	x	x				Dap-Xtra
194	1	1	runtime_cond	x		x	x	x	x	x				Dap-Xtra
195	1	1	runtime_dehum	x		x	x	x	x	x				Dap-Xtra
196	1	1	runtime_esc	x		x	x	x	x	x				Dap-Xtra
197	1	1	runtime_cwc	x		x	x	x	x	x				Dap-Xtra
198	4	20	errors	x		x	x	x	x	x				Dap-Xtra
218	6	10	errage	x		x	x	x	x	x				Dap-Xtra
258	1	1	version	x		x	x	x	x	x				Dap-Xtra
259	2	1	tmbmair	x		x	x	x	x	x				Dap-Xtra
260	2	1	tmbhum	x		x	x	x	x	x				Dap-Xtra
261	1	1	tmbairdb	x		x	x	x	x	x				Dap-Xtra
262	6	13	runtimes (as array)	x		x	x	x	x	x				Dap-Xtra
314	3	80	bitErrors	x		x	x	x	x	x				Dap-Xtra
474	4	1	adj_rate	X		X	X							Dap-Menu
478	1	1	alarm_delay_1	X		X	X							Dap-Menu
479	1	1	alarm_delay_2	X		X	X							Dap-Menu
480	1	1	alarm_delay_3	X		X	X							Dap-Menu
481	4	1	alarm_enable_1	X		X	X							Dap-Menu
482	4	1	alarm_enable_2	X		X	X							Dap-Menu
483	4	1	alarm_enable_3	X		X	X							Dap-Menu
484	4	1	alarm_select_1	X		X	X							Dap-Menu
485	4	1	alarm_select_2	X		X	X							Dap-Menu
486	4	1	alarm_select_3	X		X	X							Dap-Menu
487	1	1	chilled_water	X		X	X							Dap-Menu
488	1	1	compressor_config	X		X	X							Dap-Menu
489	1	1	c_mode	X		X	X							Dap-Menu
490	2	1	fire_lim	X		X	X							Dap-Menu
491	1	1	heat_strip_config	X		X	X							Dap-Menu
492	1	1	hi_cal	X		X	X							Dap-Menu

Array	Method	Num	Data Field	2	3	4	5	6	7	8	9	14	15	Message Type
493	1	1	hi_h_cal	X		X	X							Dap-Menu
494	2	1	hi_t_lim	X		X	X							Dap-Menu
495	1	1	humid_config	X		X	X							Dap-Menu
496	1	1	h_calib	X		X	X							Dap-Menu
497	1	1	lead_lag	X		X	X							Dap-Menu
498	1	1	loc_h_deadband	X		X	X							Dap-Menu
499	1	1	loc_h_setpt	X		X	X							Dap-Menu
500	2	1	loc_t_dband	X		X	X							Dap-Menu
501	2	1	loc_t_setpt	X		X	X							Dap-Menu
502	1	1	lo_cal	X		X	X							Dap-Menu
503	1	1	lo_h_lim	X		X	X							Dap-Menu
504	2	1	lo_t_lim	X		X	X							Dap-Menu
505	1	1	main_int	X		X	X							Dap-Menu
506	1	1	passwd_a	X		X	X							Dap-Menu
507	1	1	passwd_b	X		X	X							Dap-Menu
508	1	1	rst_mode	X		X	X							Dap-Menu
509	1	1	s_delay	X		X	X							Dap-Menu
510	2	1	t_calib	X		X	X							Dap-Menu
511	1	1	voice	X		X	X							Dap-Menu
512	1	1	vvrg	X		X	X							Dap-Menu
513	1	1	cat1	X		X	X							Dap-Menu
514	1	1	cat2	X		X	X							Dap-Menu
515	1	1	cat3	X		X	X							Dap-Menu
516	2	1	d_calib	X		X	X							Dap-Menu
517	2	1	lo_d_lim	X		X	X							Dap-Menu
518	1	1	ptc	X		X	X							Dap-Menu
519	2	1	supplyT		x	x				x				Chiller-Stat
520	2	1	returnT		x	x				x				Chiller-Stat
521	4	1	coolOn1		x	x				x				Chiller-Stat
522	4	2	coolOn2		x	x				x				Chiller-Stat
523	4	2	coolOn3		x	x				x				Chiller-Stat
524	2	1	valvePct		x	x				x				Chiller-Stat
525	4	1	pumpsOn		x	x				x				Chiller-Stat
526	4	1	condOn		x	x				x				Chiller-Stat
527	4	1	modFail		x	x				x				Chiller-Stat
528	2	1	hiSupT		x	x				x				Chiller-Stat
529	2	1	loSupT		x	x				x				Chiller-Stat
530	2	1	hiRetT		x	x				x				Chiller-Stat
531	2	1	loRetT		x	x				x				Chiller-Stat
532	1	1	csUtilPct1		x	x				x				Chiller-Stat
533	1	1	csUtilPct2		x	x				x				Chiller-Stat
534	1	1	csUtilPct3		x	x				x				Chiller-Stat

Array	Method	Num	Data Field	2	3	4	5	6	7	8	9	14	15	Message Type
535	1	1	valveUtilPct	x	x					x				Chiller-Stat
536	3	48	Errors - see Appendix E.4 for more information	x	x					x				Chiller-Stat
584	1	1	mode	x	x					x				Chiller-Stat
585	6	11	runtimes	x	x					x				Chiller-Xtra
596	3	80	errolld	x	x					x				Chiller-Xtra
676	6	10	errage	x	x					x				Chiller-Xtra
854	4	1	adjust_rate	x	x					x				Chiller-Menu
855	4	1	auto_ack	X	X					X				Chiller-Menu
856	2	1	aux_setpt	X	X					X				Chiller-Menu
857	4	1	backup_mods	X	X					X				Chiller-Menu
858	2	1	backup_setpt	X	X					X				Chiller-Menu
859	4	1	cmota	X	X					X				Chiller-Menu
860	4	1	comp_type	X	X					X				Chiller-Menu
861	4	1	ptc	X	X					X				Chiller-Menu
862	2	1	hi_r_lim	X	X					X				Chiller-Menu
863	2	1	hi_s_lim	X	X					X				Chiller-Menu
864	4	1	LL_policy	X	X					X				Chiller-Menu
865	2	1	lo_r_lim	X	X					X				Chiller-Menu
866	2	1	lo_s_lim	X	X					X				Chiller-Menu
867	1	1	main_int	X	X					X				Chiller-Menu
868	4	1	mods_configd	X	X					X				Chiller-Menu
869	1	1	network_ID	X	X					X				Chiller-Menu
870	1	1	op_1_delay	X	X					X				Chiller-Menu
871	1	1	op_2_delay	X	X					X				Chiller-Menu
872	4	1	op_1_message	X	X					X				Chiller-Menu
873	4	1	op_2_message	X	X					X				Chiller-Menu
874	1	1	password	X	X					X				Chiller-Menu
875	4	4	relay_mask_0	X	X					X				Chiller-Menu
879	4	4	relay_mask_1	X	X					X				Chiller-Menu
883	4	4	relay_mask_2	X	X					X				Chiller-Menu
887	4	1	restart_mode	X	X					X				Chiller-Menu
888	4	1	reverse_valve	X	X					X				Chiller-Menu
889	4	1	sc_alarm_on	X	X					X				Chiller-Menu
890	1	1	start_delay	X	X					X				Chiller-Menu
891	1	1	supply_dband	X	X					X				Chiller-Menu
892	2	1	supply_setpt	X	X					X				Chiller-Menu
893	4	1	temp_scale	X	X					X				Chiller-Menu
894	4	1	valve_voltage	X	X					X				Chiller-Menu
895	4	1	voice	X	X					X				Chiller-Menu
896	4	1	water_valve	X	X					X				Chiller-Menu
897	1	1	return_cal	X	X					X				Chiller-Menu

Array	Method	Num	Data Field	2	3	4	5	6	7	8	9	14	15	Message Type
898	1	1	supply_cal		X	X				X				Chiller-Menu
899	4	1	adj_rate						X	X	X	X		Dap80-Menu
900	1	1	almr_delay_1						X	X	X	X		Dap80-Menu
901	1	1	almr_delay_2						X	X	X	X		Dap80-Menu
902	1	1	almr_delay_3						X	X	X	X		Dap80-Menu
903	1	1	almr_delay_4						X	X	X	X		Dap80-Menu
904	4	1	almr_select_1						X	X	X	X		Dap80-Menu
905	4	1	almr_select_2						X	X	X	X		Dap80-Menu
906	4	1	almr_select_3						X	X	X	X		Dap80-Menu
907	4	1	almr_select_4						X	X	X	X		Dap80-Menu
908	4	1	ant-enable						X	X	X	X		Dap80-Menu
909	4	1	autoflush_time						X	X	X	X		Dap80-Menu
910	4	1	auto_ack						X	X	X	X		Dap80-Menu
911	4	1	comp_config						X	X	X	X		Dap80-Menu
912	4	1	control_type						X	X	X	X		Dap80-Menu
913	4	1	c_mode						X	X	X	X		Dap80-Menu
914	4	1	da_volts						X	X	X	X		Dap80-Menu
915	4	1	dehum_on						X	X	X	X		Dap80-Menu
916	2	1	d_calib						X	X	X	X		Dap80-Menu
917	4	1	esaver_supp_comp						X	X	X	X		Dap80-Menu
918	2	1	fire_lim						X	X	X	X		Dap80-Menu
919	4	1	heater_config						X	X	X	X		Dap80-Menu
920	2	1	hi_h_lim						X	X	X	X		Dap80-Menu
921	2	1	hi_t_lim						X	X	X	X		Dap80-Menu
922	4	1	humid_config						X	X	X	X		Dap80-Menu
923	2	1	h_calib						X	X	X	X		Dap80-Menu
924	1	1	h_dband						X	X	X	X		Dap80-Menu
925	4	1	lead_lag						X	X	X	X		Dap80-Menu
926	2	1	lo_d_lim						X	X	X	X		Dap80-Menu
927	2	1	lo_h_lim						X	X	X	X		Dap80-Menu
928	2	1	lo_t_lim						X	X	X	X		Dap80-Menu
929	2	1	main_int						X	X	X	X		Dap80-Menu
930	1	1	network_id						X	X	X	X		Dap80-Menu
931	2	1	nom_h_setpt						X	X	X	X		Dap80-Menu
932	1	1	password						X	X	X	X		Dap80-Menu
933	4	1	ptc						X	X	X	X		Dap80-Menu
934	3	16	relay_1_mask_0						X	X	X	X		Dap80-Menu
950	3	16	relay_1_mask_1						X	X	X	X		Dap80-Menu
966	3	16	relay_1_mask_2						X	X	X	X		Dap80-Menu
982	3	16	relay_2_mask_0						X	X	X	X		Dap80-Menu
998	3	16	relay_2_mask_1						X	X	X	X		Dap80-Menu
1014	3	16	relay_2_mask_2						X	X	X	X		Dap80-Menu

Array	Method	Num	Data Field	2	3	4	5	6	7	8	9	14	15	Message Type
1046	3	16	relay_3_mask_1						X	X	X	X		Dap80-Menu
1062	3	16	relay_3_mask_2						X	X	X	X		Dap80-Menu
1078	4	1	reverse_valve						X	X	X	X		Dap80-Menu
1079	4	1	rst_mode						X	X	X	X		Dap80-Menu
1080	4	1	sc_alarms						X	X	X	X		Dap80-Menu
1081	1	1	s_delay						X	X	X	X		Dap80-Menu
1082	2	1	t_calib						X	X	X	X		Dap80-Menu
1083	1	1	t_dband						X	X	X	X		Dap80-Menu
1084	2	1	t_setpt						X	X	X	X		Dap80-Menu
1085	4	1	valve_config						X	X	X	X		Dap80-Menu
1086	4	1	voice						X	X	X	X		Dap80-Menu
1100	1	1	sensor_1_name									x	x	Dap80-Analog
1101	1	1	sensor_1_units									x	x	Dap80-Analog
1102	1	1	sensor_1_type									x	x	Dap80-Analog
1103	1	1	sensor_1_min_val									x	x	Dap80-Analog
1104	1	1	sensor_1_max_val									x	x	Dap80-Analog
1105	1	1	sensor_1_cal									x	x	Dap80-Analog
1106	1	1	sensor_2_name									x	x	Dap80-Analog
1107	1	1	sensor_2_units									x	x	Dap80-Analog
1108	1	1	sensor_2_type									x	x	Dap80-Analog
1109	1	1	sensor_2_min_val									x	x	Dap80-Analog
1110	1	1	sensor_2_max_val									x	x	Dap80-Analog
1111	1	1	sensor_2_cal									x	x	Dap80-Analog
1112	1	1	sensor_1_input									x	x	Dap80-Channels
1113	1	1	sensor_2_input									x	x	Dap80-Channels
1114	1	1	sensor_3_input									x	x	Dap80-Channels

Appendix A.2. Data Table 2 - Point Descriptions

FieldServer	FieldServer Data	Point Description	Message	D/A Control Panel
1	Zone	Zone ID	DAP config	All
2	Inhibit	Unit On/ Standby/ Off	DAP config	All
3	Unit Type	Type Of DAP Panel	DAP Unit	All
4	temp	Current Temperature	DAP Stat	All
5	hum	Current Humidity	DAP Stat	All
6	d-temp	Current discharge	DAP Stat	All
7	mode	Current Unit mode (see	DAP Stat	All
23	hold	Current Functions are	DAP Stat	All
39	cs_on	Cooling stages on	DAP Stat	All
40	hs_on	Heating stage on	DAP Stat	All

FieldServer	FieldServer Data	Point Description	Message	D/A Control Panel
41	valvePCT	Opening Percentage of	DAP Stat	All
42	hVlvPCT	Opening Percentage of	DAP Stat	All
43	errors	Alarm (see note 3)	DAP Stat	All
57	hiTemp	Highest Temperature in	DAP Stat	All
58	loTemp	Lowest Temperature in	DAP Stat	All
59	hiHum	Highest Humidity in the	DAP Stat	All
60	LoHum	Lowest Humidity in the	DAP Stat	All
61	Chilled_water	Water Valve setting	DAP Stat	All
62	Compressor_config	Compressor	DAP Stat	All
63	heat_strip_config	Heating configuration	DAP Stat	All
64	hum_config	Humidifier configuration	DAP Stat	All
65	csUtilPct	Utilization Percentage of	DAP Stat	All
66	hsUtilPct	Utilization Percentage of	DAP Stat	All
67	ValveUtilPct	Utilization Percentage of	DAP Stat	All
68	humUtilPct	Utilization Percentage of	DAP Stat	All
69	alarm_select_1	Message for optional	DAP Stat	All
70	alarm_select_2	Message for optional	DAP Stat	All
71	alarm_select_3	Message for optional	DAP Stat	All
72	alarm_select_4	Message for optional	DAP Stat	All
73	bitErrors	Alarm History	DAP Stat	All
185	runtime_c1	Compressor 1 runtime	DAP-Xtra	All
186	runtime_c2	Compressor 2 runtime	DAP-Xtra	All
187	runtime_c3	Compressor 3 runtime	DAP-Xtra	DAP046 only
188	runtime_c4	Compressor 4 runtime	DAP-Xtra	DAP046 only
189	runtime_ht1	Heating stage 1 runtime	DAP-Xtra	All
190	runtime_ht2	Heating stage 2 runtime	DAP-Xtra	All
191	runtime_ht3	Heating stage 3 runtime	DAP-Xtra	All
192	runtime_hum	Humidification runtime	DAP-Xtra	All
193	runtime_evap	Blower/ Fan runtime	DAP-Xtra	All
194	runtime_cond	Condenser runtime	DAP-Xtra	All
195	runtime_dehum	dehumidification runtime	DAP-Xtra	All
196	runtime_esc	Energy saver runtime	DAP-Xtra	All
197	runtime_cwc	Chilled water Valve	DAP-Xtra	All
198	errors	Alarm History	DAP-Xtra	All
218	errage	Alarm History time (hrs)	DAP-Xtra	All
258	version	DAP II software revision	DAP-Xtra	All
259	tmbmair	Temperature setpoint	DAP-Xtra	DAP049,048,046
260	tmbhum	Humidity setpoint	DAP-Xtra	DAP049,048,046
261	tmbairdb	Temperature deadband	DAP-Xtra	DAP049,048,046
262	runtimes	runtimes array (in hrs)	DAP-Xtra	DAP049,048,046
314	bitErrors		DAP-Xtra	DAP049,048,046
474	adj_rate	Adjustment rate	DAP Menu	DAP049,048,046

FieldServer	FieldServer Data	Point Description	Message	D/A Control Panel
478	alarm_delay_1	optional alarm 1 delay	DAP Menu	DAP049,048,046
479	alarm_delay_2	optional alarm 2 delay	DAP Menu	DAP049,048,046
480	alarm_delay_3	optional alarm 3 delay	DAP Menu	DAP049,048,046
481	alarm_enable_1	Enable optional alarm 1	DAP Menu	DAP049,048,046
482	alarm_enable_2	Enable optional alarm 2	DAP Menu	DAP049,048,046
483	alarm_enable_3	Enable optional alarm 3	DAP Menu	DAP049,048,046
484	alarm_select_1	Select alarm message	DAP Menu	DAP049,048,046
485	alarm_select_2	Select alarm message	DAP Menu	DAP049,048,046
486	alarm_select_3	Select alarm message	DAP Menu	DAP049,048,046
487	Chilled_water	Chilled water Valve	DAP Menu	DAP049,048,046
488	Compressor_config	Compressor	DAP Menu	DAP049,048,046
489	c_mode	Unit in Centigrade	DAP Menu	DAP049,048,046
490	fire_lim	Firestat temperature	DAP Menu	DAP049,048,046
491	heat_strip_config	Heating configuration	DAP Menu	DAP049,048,046
492	hi_cal	A to D high limit setting	DAP Menu	DAP049,048,046
493	hi_h_limit	High humidity alarm limit	DAP Menu	DAP049,048,046
494	hi_t_lim	High temperature alarm	DAP Menu	DAP049,048,046
495	humid_config	Humidifer configuration	DAP Menu	DAP049,048,046
496	h_calib	Humidity calibration	DAP Menu	DAP049,048,046
497	lead_lag	Compressor lead/lag	DAP Menu	DAP049,048,046
498	loc_h_ddband	humidity deadband	DAP Menu	DAP049,048,046
499	loc_h_setpt	Humidity setpoint	DAP Menu	DAP049,048,046
500	loc_t_dband	Temperature deadband	DAP Menu	DAP049,048,046
501	loc_t_setpt	Temperature setpoint	DAP Menu	DAP049,048,046
502	lo_cal	A to D low limit setting	DAP Menu	DAP049,048,046
503	lo_h_lim	Low humidity alarm limit	DAP Menu	DAP049,048,046
504	lo_t_lim	Low temperature alarm	DAP Menu	DAP049,048,046
505	main_int	Maintenance schedule	DAP Menu	DAP049,048,046
506	passwd_a	Password A setting	DAP Menu	DAP049,048,046
507	passwd_b	Password B setting	DAP Menu	DAP049,048,046
508	rst_mode	Restart mode setting	DAP Menu	DAP049,048,046
509	s_delay	Start delay setting (in	DAP Menu	DAP049,048,046
510	t_calib	Temperature calibration	DAP Menu	DAP049,048,046
511	voice	Audible alarm tone	DAP Menu	DAP049,048,046
512	vvrg	Chilled water output	DAP Menu	DAP049,048,046
513	cat1	Alarm relay #1 category	DAP Menu	DAP 046
514	cat2	Alarm relay #2 category	DAP Menu	DAP 046
515	cat3	Alarm relay #3 category	DAP Menu	DAP 046
516	d_calib	Discharge Air sensor	DAP Menu	DAP 046
517	lo_d_lim	Low Discharge	DAP Menu	DAP 046
518	ptc	Person to contact on	DAP Menu	DAP 046
All the points of Chiller Panel are omitted from this list				

FieldServer	FieldServer Data	Point Description	Message	D/A Control Panel
899	adj_rate	Adjustment rate setting	DAP80 Menu	DAP80
900	alarm_delay_1	optional alarm 1 delay	DAP80 Menu	DAP80
901	alarm_delay_2	optional alarm 2 delay	DAP80 Menu	DAP80
902	alarm_delay_3	optional alarm 3 delay	DAP80 Menu	DAP80
903	alarm_delay_4	optional alarm 4 delay	DAP80 Menu	DAP80
904	alarm_select_1	Select alarm message	DAP80 Menu	DAP80
905	alarm_select_2	Select alarm message	DAP80 Menu	DAP80
906	alarm_select_3	Select alarm message	DAP80 Menu	DAP80
907	alarm_select_4	Select alarm message	DAP80 Menu	DAP80
908	ant_enable	Humidity anticipation	DAP80 Menu	DAP80
909	autoflush_time	Autoflush timer setting	DAP80 Menu	DAP80
910	auto_ack	Automatic self-test	DAP80 Menu	DAP80
911	Comp_config	Compressor	DAP80 Menu	DAP80
912	control_type	Control type setting	DAP80 Menu	DAP80
913	c_mode	display temperature	DAP80 Menu	DAP80
914	DA_volts	Water Valve voltage	DAP80 Menu	DAP80
915	dehum_on	Dehumidification mode	DAP80 Menu	DAP80
916	d_calib	Discharge Air sensor	DAP80 Menu	DAP80
917	esaver_supp_comp	Energy saver	DAP80 Menu	DAP80
918	fire_lim	Firestat temperature limit	DAP80 Menu	DAP80
919	heater_config	Heating configuration	DAP80 Menu	DAP80
920	hi_h_lim	High humidity alarm limit	DAP80 Menu	DAP80
921	hi_t_lim	High temperature alarm	DAP80 Menu	DAP80
922	humid_config	Humidifier configuration	DAP80 Menu	DAP80
923	h_calib	Humidity calibration	DAP80 Menu	DAP80
924	h_dband	humidity deadband	DAP80 Menu	DAP80
925	lead_lag	Compressor lead/lag	DAP80 Menu	DAP80
926	lo_d_lim	Low Discharge	DAP80 Menu	DAP80
927	lo_h_lim	Low humidity alarm limit	DAP80 Menu	DAP80
928	lo_t_lim	Low temperature alarm	DAP80 Menu	DAP80
929	main_int	Maintenance schedule	DAP80 Menu	DAP80
930	network_id	set unit Network ID	DAP80 Menu	DAP80
931	nom_h_setpt	Humidity setpoint	DAP80 Menu	DAP80
932	password	password	DAP80 Menu	DAP80
933	ptc	Person to contact on	DAP80 Menu	DAP80
934	relay_1_mask_0	selection for alarm	DAP80 Menu	DAP80
950	relay_1_mask_1	selection for alarm	DAP80 Menu	DAP80
966	relay_1_mask_2	selection for alarm	DAP80 Menu	DAP80
982	relay_2_mask_0	selection for alarm	DAP80 Menu	DAP80
998	relay_2_mask_1	selection for alarm	DAP80 Menu	DAP80
1014	relay_2_mask_2	selection for alarm	DAP80 Menu	DAP80
1030	relay_3_mask_0	selection for alarm	DAP80 Menu	DAP80

FieldServer	FieldServer Data	Point Description	Message	D/A Control Panel
1046	relay_3_mask_1	selection for alarm	DAP80 Menu	DAP80
1062	relay_3_mask_2	selection for alarm	DAP80 Menu	DAP80
1078	reverse_valve	Chilled water direction	DAP80 Menu	DAP80
1079	rst_mode	Restart mode setting	DAP80 Menu	DAP80
1080	sc_alarms	Compressor short cycle	DAP80 Menu	DAP80
1081	s_delay	Start delay setting (in	DAP80 Menu	DAP80
1082	t_calib	Temperature calibration	DAP80 Menu	DAP80
1083	t_dband	Temperature deadband	DAP80 Menu	DAP80
1084	t_setpt	Temperature setpoint	DAP80 Menu	DAP80
1085	Valve_config	Chilled water Valve	DAP80 Menu	DAP80
1086	voice	Audible alarm tone	DAP80 Menu	DAP80
1100	sensor_1_name	Optional Analog sensor 1	DAP80 Menu	DAP80 W/analog
1101	sensor_1_units	Optional Analog sensor 1	DAP80 Menu	DAP80 W/analog
1102	sensor_1_type	Optional Analog sensor 1	DAP80 Menu	DAP80 W/analog
1103	sensor_1_min_val	Optional Analog sensor 1	DAP80 Menu	DAP80 W/analog
1104	sensor_1_max_val	Optional Analog sensor 1	DAP80 Menu	DAP80 W/analog
1105	sensor_1_cal	Optional Analog sensor 1	DAP80 Menu	DAP80 W/analog
1106	sensor_2_name	Optional Analog sensor 2	DAP80 Menu	DAP80 W/analog
1107	sensor_2_units	Optional Analog sensor 2	DAP80 Menu	DAP80 W/analog
1108	sensor_2_type	Optional Analog sensor 2	DAP80 Menu	DAP80 W/analog
1109	sensor_2_min_val	Optional Analog sensor 2	DAP80 Menu	DAP80 W/analog
1110	sensor_2_max_val	Optional Analog sensor 2	DAP80 Menu	DAP80 W/analog
1111	sensor_2_cal	Optional Analog sensor 2	DAP80 Menu	DAP80 W/analog
1112	sensor_1_input	sensor 1 input	DAP80 Menu	DAP80 W/analog
1113	sensor_2_input	sensor 2 input	DAP80 Menu	DAP80 W/analog
1114	sensor_3_input	sensor 3 input	DAP80 Menu	DAP80 W/analog

Note 1: mode: 2' bit definitions:		Note 2: hold: 2' bit definitions:	
0X01	Unit is in centigrade mode	0X01	needed cooling is inhibited
0X02	RFU , always 1	0X02	needed heatingg is inhibited
0X04	RFU , always 1	0X04	needed humidification is inhibited
0X08	Unit is dehumidifying	0X08	needed dehumidification is inhibited
0X10	Unit is humidifying	0X10	not used
0X20	Energy save is active	0X20	not used
0X40	not used	0X40	Network"standby" inhibit is active
0X80	not used	0X80	Network"off" inhibit is active
Note 2: errors: 7*2' bit definitions:			
00	Manual override:check bypass switches		errors[0], bit 0x01
01	water detected under floor		errors[0], bit 0x02
02	No air flow, check belt and motor		errors[0], bit 0x04
03	dirty filter: check filter		errors[0], bit 0x08

04	Humidifier problem: check water pressure	errors[0], bit 0x10
05	Low voltage warning	errors[0], bit 0x20
06	Firestat alarm, unit shutdown	errors[0], bit 0x40
07	compressor short cycle	errors[0], bit 0x80
08	power problem or unit restart	errors[1], bit 0x01
09	humidity sensor problem	errors[1], bit 0x02
10	temperature sensor problem	errors[1], bit 0x04
11	schedule maintenance due	errors[1], bit 0x08
12	high presure C1: manual reset	errors[1], bit 0x10
13	low presure C1: auto reset	errors[1], bit 0x20
14	high presure C2: manual reset	errors[1], bit 0x40
15	low presure C2: auto reset	errors[1], bit 0x80
16	smoke detector: unit shutdown	errors[2], bit 0x01
17	No water flow :check pump	errors[2], bit 0x02
18	Discharge temperature sensor problem	errors[2], bit 0x04
19	High Temperature warning	errors[2], bit 0x08
20	Low temperature warning	errors[2], bit 0x10
21	High humidity warning	errors[2], bit 0x20
22	low humidity warning	errors[2], bit 0x40
23	Fan motor overload: check motor amperage	errors[2], bit 0x80
24	Local alarm 1: see tag inside door	errors[3], bit 0x01
25	Local alarm 2: see tag inside door	errors[3], bit 0x02
26	Local alarm 3: see tag inside door	errors[3], bit 0x04
27	Local alarm 4: see tag inside door	errors[3], bit 0x08
28	Standby Pump on: check primary pump	errors[3], bit 0x10
29	UPS power on: check primary power	errors[3], bit 0x20
30	Custom message on optional input 1	errors[3], bit 0x40
31	Custom message on optional input 2	errors[3], bit 0x80
32	Custom message on optional input 3	errors[4], bit 0x01
33	Custom message on optional input 4	errors[4], bit 0x02
34	Humidification inhibited	errors[4], bit 0x04
35	Reheat inhibited	errors[4], bit 0x08
36	Reheat and humidification inhibited	errors[4], bit 0x10
37	Discharge air temperature limit	errors[4], bit 0x20
38	Reheat mode during dehumidication	errors[4], bit 0x40
39	Manual override:check bypass switches	errors[4], bit 0x80
40	High condensate water level	errors[5], bit 0x01

Appendix A.3. Data Table 3 - Unit Types

When the driver reads everything from a device it must first obtain the device's unit type so that it can determine what other data is available. Once the unit type is obtained then the driver updates the 'Unit-Type' field visible on the node screen of the RUIDebug program. The unit type is also available in the Data Array defined in table 4.4.5.1

The following table lists the unit types that can be processed by this driver.

Unit Type	Numeric Unit Type	Description
"-"	0	Unknown/unavailable/un-initialized
"1"	1	044 data logger
"2"	2	046 expanded DAP
"3"	3	046 2 mod chiller
"4"	4	046 3 mod chiller
"5"	5	048 DAP, 80-character display
"6"	6	049 DAP, 16-character display
"7"	7	080 DAP II, no relay expansion
"8"	8	080 Chiller II
"9"	9	080 DAP II, with relay expansion
"10"	10	Not Defined
"11"	11	Not Defined
"12"	12	Not Defined
"13"	13	Not Defined
"E"	14	080 DAP II, with analog module
"F"	15	080 DAP II, with relay and analog

Appendix B. ADVANCED TOPICS

Appendix B.1. Additional Driver Specific Map Descriptor Parameters

The driver offers advanced configuration by adding to and extending the Map Descriptors specific to the Dart Serial Driver.

Column Title	Function	Legal Values
DA_Func*	Specifies the Data Aire Command/Query function to be used. Use a function appropriate to the type of Server (DAP/Chiller/DAP80) and the type of data required.	Numeric/Text. See Appendix B.2 for a list of possible values.
DA_Field*	Specifies the data field to be retrieved from the Server device. Servers are only capable of responding with a data composite consisting of many data fields. This parameter is used to specify the parameters to be extracted from the data composite. Note1.	Text. See Appendix B.3 for a list of possible values as well as Table 4.4.5.1
DA_Assoc*	This field is used to make the association between passive and active (rdbc/wrbc) Map Descriptors addressing the same node. Give the rdbc and its associated passive Map Descriptors the same value (any number) and give the wrbc and its passive Map Descriptors another value for DA_Assoc.	Any positive integer.
Da_Freq	Used only for connection to DART's. Specifies in milliseconds the intervals between wrbc/rdbc Map Descriptor execution.	> 18000 (3 minutes) 30000 (5 minutes) recommended
DA_Method [‡]	Specifies the extraction method. Such as Hex-ASCII to decimal number in 10's of a degree,	See Appendix B.4 for a list of possible values.
DA_Bytcnt [‡]	Specifies the number of bytes that are to be processed by the method specified above. For method#6 which processes an array of elements the DA_Bytcnt specifies the number of bytes that constitute each element of the array.	>= 1
DA_Offset [‡]	An offset into the data composite that is returned when the Server is polled. The offset is the number of bytes from the first data byte.	0 to the length of the data composite. No validation is performed.
DA_Elec [‡]	Number of elements that are produced by the extraction method.	>= 1

[‡] These parameters are only required for custom data extractions not provided for with DA_Field parameter.

Appendix B.2. DA_Func Parameter - Permitted values.

The driver supports a limited subset of the Dart Poll & Response Functions. The selection of the sub-set is based on the identification of useful & practical functions.

In addition to the 'Everything' keyword indicated in chapter 4 the following specific query functions are implemented.

Func.	Description	Driver Parameter	Protocol Id.
'1'	DART Config Query	DA_Func = dart-config	49
'2'	Dart Psswd Query	DA_Func = dart-password	50
'3'	DAP Config Command	DA_Func = dap-config	51
'4'	DAP Log Query	DA_Func = dap-log	52
'5'	DAP Unit-Type Query	DA_Func = dap-unit	53
'6'	DAP Stat Query	DA_Func = dap-stat	54
'7'	DAP Xtra Query	DA_Func = dap-xtra	55
'8'	DAP Menu Query	DA_Func = dap-menu	56
'A'	Chiller Stat Query	DA_Func = chiller-stat	65
'B'	Chiller Xtra Query	DA_Func = chiller-xtra	66
'C'	Chiller Menu Query	DA_Func = chiller-menu	67
'D'	Dart Status	DA_Func = dart-status	68
'E'	DAP80 Menu Query	DA_Func = dap80-menu	69
'G'	DAP Analog Query	DA_Func = dap80-analog	71
'H'	DAP Channels Query	DA_Func = dap80_channles	72

Each of the above queries returns a complex set of data consisting of many sub-fields. Contact Data-Aire for a complete listing of the data composite returned. The following special / diagnostic functions are also implemented.

Driver Parameter	Protocol Id.
DA_Func = All-Listen	11
DA_Func = Ack	6
DA_Func = Dart-Transparant	2
DA_Func = Dart-Opaque	3
DA_Func = Test-Echo	16
DA_Func = Test-No-Echo	15
DA_Func = Unit-Talk	13

With the exception of the Unit-Talk transmission of this command, these are nodeless commands to allow the units time to switch their mechanical command. When using any of these special commands no other DA_* fields need be specified.

All-Listen instructs the all units in the network to switch their relays to the listen position. Those units already in the listen position will do nothing. Those in the talk position will first echo the all-listen command and then switch their relays to the listen position. A pause of 0.15 seconds is required after the relays.

Appendix B.3. DA_Field Parameter - Permitted values⁴.

DA_Field Legal Values	Description		
All	<p>The whole data record returned by the Server is stored in the Data Array byte for byte. The number of bytes written is dependent on the DA_Func.</p>	DA_Func	# Bytes
		DA_Func=dart-password	231
		DA_Func=dart-config	41
		DA_Func=dap-config	4
		DA_Func=dap-log	240
		DA_Func=dap-unit	1
		DA_Func=dap-stat	68
		DA_Func=dap-xtra	124
		DA_Func=dap-menu	103
		DA_Func=chiller-stat	54
		DA_Func=chiller-xtra	104
		DA_Func=chiller-menu	89
		DA_Func=dap80-menu	138
	DA_Func=dart-status	9	
	DA_Func=dap80-analog	36	
	DA_Func=dap80-channels	12	
Special	<p>Indicates that a user defined extraction is specified in the Map Descriptor. When this value is specified as the DA_Field value then DA_Method, DA_Bytcnt, DA_Offset, DA_Elecnt must also be specified.</p>		

⁴ See Section Appendix A for all other keywords.

Appendix B.4. DA_Method Parameter Values and Notes

The DA_Method specifies a method for interpreting a range of bytes when the DA_Field=special.

Method #	Description
Method 1:	<p>Each byte is valid when it contains only one of the following ASCII characters. { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F }</p> <p>Each byte being parsed is considered to be a hexadecimal digit with the most significant digit being the left-most byte.</p> <p>The four bytes 30 31 32 33 (hex) are therefore interpreted as the hexadecimal number 0123 and the decimal value is equal to 291.</p>
Method 2	<p>As method 1, but used for humidity's and temperatures which are transmitted as the number of tenths of a unit – Thus the 4 bytes yield the decimal number 29.1 °F/%.</p>
Method 3	<p>Each byte is regarded as containing a hexadecimal digit in ASCII format.</p> <p>Example: incoming byte contains 41(hex). -> regarded as the hexadecimal digit 'A' in ASCII format.</p> <p>The hex digit is converted to a series of 8 bits. In this example the bits are 00001010 with the msb being the left-most.</p>
Method 4	<p>As Method 3 except that the decimal value of this digit is written to the Data Array. In this example the number 10 would be written to the Data Array.</p>
Method 5	<p>There is no translation - the raw bytes are written to the data array.</p>
Method 6	<p>Processes an array of elements using method 1 translation. The raw data being parsed is considered to consist of DA_elecnt elements each consisting of DA_bytecnt bytes.</p>

Appendix B.5. Advanced Map Descriptor 1:Collecting Data From Multiple Fileds

```
// Client Side Nap Descriptors
```

Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Function	Node_name	Address	Length	DA_Func	DA_Field	Scan_Interval
A1	, DA_AI3	, 0	, RDBC	, Node_A	, 0	, 1	, dap-stat	, temp	, 5
A2	, DA_AI3	, 1	, passive	, Node_A	, 0	, 1	, dap-stat	, hum	, 5
A3	, DA_AI3	, 2	, passive	, Node_A	, 0	, 1	, dap-stat	, d_temp	, 5
A4	, DA_AI3	, 3	, passive	, Node_A	, 0	, 1	, dap-stat	, hiTemp	, 5
A5	, DA_AI3	, 4	, passive	, Node_A	, 0	, 1	, dap-stat	, loTemp	, 5
A6	, DA_AI2	, 0	, passive	, Node_A	, 0	, 1	, dap-stat	, cs_on	, 5
A7	, DA_AI2	, 1	, passive	, Node_A	, 0	, 1	, dap-stat	, hs_on	, 5

It would be sensible for DA_AI3 to be an array of FLOATs because the temps and humidity's return real numbers with one digit after the decimal point.

DA_AI2 could be any type of array other than BIT because the values returned for these parameters are whole numbers less than 255.

All these Map Descriptors address Node_A therefore only one Map Descriptor needs to read (rdbc) the node. The remaining Map Descriptors can be passive (thus optimizing communications.)

All these Map Descriptors read their data from the same Server

Server is a DAP reading status information.

The scan time is only important for the active Map Descriptor.

These parameters must be typed in exactly as specified in this manual. They are case sensitive. The format of the data extracted depends on the parameter.

Appendix B.6. Advanced Map Descriptor Example 2:Collecting Error Messages

The DAP-II Status query returns 14 bytes of errors & status information. The arrangement and meaning of these bytes is defined by the Data Aire Corporation and is also dependent on the type of DAPII module being polled.

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

Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Function	Node_name	Address	Length	DA_Func	DA_Field	Scan_Interval
A1	DA_AI1	0	RDBC	Node_A	0	1	dap-stat	errors	5
A2	DA_AI1	1	passive	Node_A	0	1	dap-stat	bitErrors	5

The 'errors' key word returns 14 bytes, thus the DA_AI1 should be a BYTE array. Each byte will have values 0-15 to represent the value of the bits in each byte.

The bitErrors extracts the same data from the DAP but presents it as a series of 14x8 bits. Thus DA_AI1 should be a BIT array

Data arrangement and meaning of each error is defined by Data Aire Corp.

Example: Bit 24 is a LOW TEMP WARNING for DAPII-044/8/9 units.

Example: Bit 09 is a HUMIDITY SENSOR PROBLEM for a DAPII-080 unit

Server is a DAP and is reading status information.

BitErrors is a synonym for errors. The data is extracted using a different data format.

Appendix B.7. Advanced Map Descriptor Example 3: Using Special DA Field

```
// Client Side Map Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_name , Address , Length , DA_Func, DA_Field , DA_Method , DA_Bytecnt , DA_Offset , DA_Elecnt , Scan_Interval
A1 , DA_AI3 , 0 , RDBC , Node_A , 0 , 1 , dap-stat , special , 1 , 4 , 10 , 1 , 5
```

It would be sensible for DA_AI3 to be an array of Floats because extraction method#1 returns a floating point number.

Specials can be used as rdbc and passive Map Descriptors.

Performs a DAP status query

If parameter 'special' is used, additional parameters must be specified.

See Appendix B.4 for descriptions on how these extraction methods work.

From the data bytes returned by the Server, extract 4 bytes starting at byte 10 and apply method 1 to convert the bytes before writing them to the FieldServer Data Array.

Appendix B.8. Advanced Map Descriptor Example 4: - Using DA_Assoc

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

Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Function	Node_name	Address	Length	DA_Func	DA_Field	DA_Assoc	Scan_Interval
A1	, DA_AI3	, 0	, RDBC	, Node_A	, 0	, 1	, dap80-menu	, All	, 1	, 5
A2	, DA_AI3	, 1	, passive	, Node_A	, 0	, 1	, dap80-menu	, alm_delay_1	, 1	, 5
A3	, DA_AI3	, 2	, passive	, Node_A	, 0	, 1	, dap80-menu	, alm_delay_2	, 1	, 5
A4	, DA_AI3	, 3	, passive	, Node_A	, 0	, 1	, dap80-menu	, alm_delay_3	, 1	, 5
A8	, DA_AI4	, 0	, WRBC	, Node_A	, 0	, 1	, dap80-menu	, nom_h_setpt	, 2	, 5
A9	, DA_AI4	, 0	, passive	, Node_A	, 0	, 1	, dap80-menu	, t_setpt	, 2	, 5

A 'Read' and some passive Map Descriptors to extract other data fields from the same read. (optimizes communications)

Read must precede the passive's.

A 'Write' and a passive to the same node. This write will update two fields, the nom_h_setpt and the t_setpt.

The write Map Descriptor. Must precede the passive.

Potential confusion for the FieldServer because the node and the DA_Func's are the same for all the Map Descriptors.

Solve this problem using DA_Assoc.

DA_Assoc associates the passives with the correct active Map Descriptor. Map Descriptors A2, 3, 4 are associated with A1 by the value of DA_Assoc=1

Map Descriptor A9 is associated with A8 by the value of DA_Assoc=2

Appendix B.9. Advanced Map Descriptor: Example 5 - Using a special / diagnostic command.:

```
// Client Side Map Descriptors

Map_Descriptor_Name      , Data_Array_Name      , Data_Array_Offset      , Function      , Node_Name      , Address      , Length      , DA_Func      , Scan_Interval
A1                        , UNUSED_ARRAY      , 0                      , WRB           , No_Node        , 0            , 1           , All-Listen   , 5
```

A Data Array must be associated with the Map Descriptor even though it will not be used. It may be any data type.

This command is sent only once. To do this periodically change this to wrbc.

Must connect this Map Descriptor to a node whose Node_ID is zero. For example.

```
Nodes
Node_Name  , Node_ID  , Protocol  , Port
Unit1      , 0        , Daire     , R1
```

This is a special / diagnostic command. It causes a one byte message to be sent.

Appendix B.10. Advanced Map Descriptor: Example 6 – Turning Dap's On/Off.

This Map Descriptor sends a Dap-Config command to the DAP device. The Map Descriptor extracts two consecutive values from the Data Array the first value being the zone number and the second the inhibit value. Additional information is provided in 0.

```
// Client Side Map Descriptors

Map_Descriptor_Name      , Data_Array_Name      , Data_Array_Offset      , Function      , Node_name      , Address      , Length      , DA_Func      , Scan_Interval
INH_04,                  , DA_WR_01             , 0              , WRBC           , Node_A       , 0            , 2            , Dap-Config    , 5
```

Two elements of this array are used. The value found at offset 0 is the zone number. The value found at offset 1 is the inhibit code which is used to inhibit/run units.

Appendix C. TROUBLESHOOTING TIPS

Appendix C.1. Bad Values

In the event that the driver cannot correctly decode the raw bytes it will generally write a value which indicates bad data. In most cases the indicating value is -1 or 65535 (depending on data type). When setting bits for status fields the driver will not write new data to the array if the incoming byte is invalid. Consult the error log for indication of this type of problem.

Example: Valid ASCII digits are 0...9, A...F. If a byte is being parsed and a hex digit is expected but not found then the driver considers this an error and writes the bad value indication **OR** produces an error message when the bad value indication cannot be used.

Appendix C.2. Dead Nodes

When a node is absent or dies, the driver may go idle and stop communicating with all other nodes. This problem is specific to the Data Aire devices.

Appendix C.3. Noise

A high percentage of responses on the secondary port are preceded with noise. The driver is generally able to filter this noise because legitimate characters in a response are limited to the ASCII character set and a few control characters. The driver does not filter noise in the middle of a message. Thus the message will fail (generally as a checksum failure) when there is noise in the middle of a message. This is normal for most drivers.

Appendix C.4. Ignored Messages

Ignored messages which are messages sent by a DAP/DART for which the driver cannot find an appropriate Map Descriptor are reported by the driver. This simply indicates that a message containing information that is not required is being discarded. The current version of the driver ignores a few messages relating to the status of the DART device. Later versions of the driver will be capable of storing these messages and the number of ignored messages will decrease.

Appendix C.5. Driver limitation

Unable to reset run times.

Appendix C.6. Baud Rate.

Rev 9.5 of the DAP firmware changed the baud rate from 2400 to 9600. All units must have the same baud rate on the network. The customer should check the firmware version of each unit and record these in a table. FieldServer recommends that all units have the same version due to the fact that instability between different versions has been experienced.

Appendix C.7. Wiring.

This must be completed exactly as specified by Data Aire. Variations in the wiring invariably causes problems. The wiring/installation guide is available from Data Aire.

Appendix C.8. Stuck Unit.

Communications on the DAP units requires the operation of a mechanical relay. This can get stuck. The Front Panel menu of each DAP unit allows a user to execute a diagnostic test of the comms board. The customer should execute this test and use the Data Aire installation guide to resolve any issues. Typically a failed unit requires replacement. A single failed card will fail the whole network. The FieldServer sends polls on the primary port and expects responses on the secondary port. The fact that we get back a copy of the poll message on the secondary port does not mean that the units are operating correctly since this is possible when the units are stuck in pass thru mode.

Appendix C.9. Number of Units

Up to 250 DAP devices can be connected to a FieldServer using the DAP driver.

Normal RS485 network considerations apply - these limit the number of units based on the load of each node.

- For unit loads only 32 devices per network can be used.
- For .25 unit loads 128 devices can be used on one network segment. If more than 32 full unit loads are required then the network must be segmented and connected via a repeater
- It may be necessary to use a repeater if total RS-485 length exceeds 1000ft.

It is recommended that tests are performed with a single or few units.

Appendix D. ERROR MESSAGES

Message	Description	Discussion
DAP:#1	FYI. The Map Descriptor called <%s> is too short.	The Map Descriptor used to expose driver stats must be 500 elements long. Change the length parameter and ensure that the Data Array is long enough too ⁵
DAP:#2	FYI. You could have used a Map Descriptor called <%s> to expose diagnostic info.	This message is for information only and may be ignored. The driver is capable of exposing performance and communication statistics. See Appendix D.1 for more information
DAP:#3	Err. Method requires even number of bytes. nb= %d Map Descriptor= <%s>	May be a corrupted message. If frequently occurring, take a log and report to FieldServer Technical Services.
DAP:#4a	Err. Method3 is translating invalid bytes. ch1= %d(dec) %d(dec)	The protocol only allows certain characters to be transmitted. If an unexpected character is encountered then the error message is printed. If the error occurs frequently the make a log file and report the error to FST. The error may arise from a corrupted message.
DAP:#4b	Err: Method2 is translating invalid bytes. ch= %d(dec) %d(dec)	
DAP:#4c	Err: Method1 is translating invalid bytes. ch= %d(dec) %d(dec)	
DAP:#5	Err. Method3 is translating invalid bytes. ch2= %d(dec) %d(dec)	
DAP:#6	Err. Invalid Station (Dec) %d-%d-%d	Only certain station numbers are legal. (1-260). If the error occurs frequently then make a log file and report the error to FST. The error may arise from a corrupted message.
DAP:#7	Err: Station= %d illegal - forcing to 1	The CSV file contains a DAP node with an illegal station number. The driver has changed the illegal node number to 1. It is important that this error is corrected or data will be collected from node1.5
DAP:#8	Err. secondary_port must be defined.	The DAP driver sends polls on one port and expects responses on a second port, therefore 2 ports must be defined for each DAP network – see Section Error! Reference source not found.5
DAP:#9	FYI. Driver on port= %d suitable for NON-Dart applications only.	This message is for information only and may be ignored. The message indicates that the configuration is suitable for connection to a DAP only network. No DART device may be present in the network.
DAP:#10	Err. Invalid Station#(%d) . (1-260) Map Descriptor= <%s>	The CSV file contains a DAP node with an illegal station number. Only station numbers between 1 & 260 are permitted. ⁵
DAP:#11	Err. Invalid Daire function (%d) .Map Descriptor= <%s>	The value of the DA_Function parameter specified in the CSV file cannot be recognized by the driver. Read Appendix B.2 for more information. ⁶
DAP:#12	Err: Invalid Daire field (%d) .Map Descriptor= <%s>	
DAP:#13	Err. Invalid Daire method (%d) .Map Descriptor= <%s>	The value of the DA_Method parameter specified in the CSV file cannot be recognized by the driver. Read Appendix B.4 for more information. ⁶
DAP:#14	Err. Map Desc. Cant understand function= %d <%s>	The value of the DA_Function parameter specified in the CSV file cannot be recognized by the driver. Read Appendix B.2 for more

⁵ Edit the CSV file and reset the FieldServer for the changes to take effect.

⁶ Edit the CSV file and reset the FieldServer for the changes to take effect.

Message	Description	Discussion
		information.6
DAP:#15	Err. Map Desc. Cant understand id= %d <%s>	The value of the DA_Field parameter specified in the CSV file cannot be recognized by the driver. Read Appendix B.3 for more information. 6
DAP:#16	Err. Unknown Response= %X(h)	A message has been received that cannot be recognized. If this error occurs repeatedly, make a log file and report the error to FST. If the error occurs occasionally it may be indicating an occasional corrupt message and may be ignored.
DAP:#17	Err. Unknown Query	
DAP:#18	Err. Unknown Command	
DAP:#19	Err. Unknown Msg Type	
DAP:#20	Writing before Unit Type is known.	If you see this message printed report the fact to FST immediately.
DAP:#21	Err. Cant write this function= %c. Map Descriptor= <%s>	Only certain data is writable. See Section Appendix A.2 for more information.5
DAP:#22	FYI. Write abandoned. Map Descriptor=<%s>. ch=%d j=%d off=%d dev=<%s>	It is not possible to write to a DAP device before it has been read successfully. This message arises if the device has been read but some data contained in the read is invalid. The driver is preventing the invalid data being written back to the device. If this message only occurs once or twice after the FieldServer is restarted, it may be ignored. If the message occurs repeatedly then take a log file and report the error to FST
DAP:#23 DAP:#23a	FYI. Write abandoned. Read incomplete. Will retry. Map Descriptor= <%s>	It isn't possible to write to a DAP device before it has been read successfully. If this message only occurs once or twice after the FieldServer is restarted, it may safely be treated as for information only. If the message occurs repeatedly then the configuration may need changing.
DAP:#24	Err. Rev-Translate: Unknown Method.	Report this error to FST providing a copy of your CSV file.
DAP:#25	Err. Field= <%s> cannot be used for write. Map Descriptor= <%s>	This error arises when (1) a Map Descriptor refers to a field that is not unique but the function has not been specified or (2) when a non-writable field has been specified. See section 4.4.5 for additional information. This error could also be produced if there are two (or more) write Map Descriptors each having a different DA_Func but with the DA_Assoc values of each Map Descriptor set equal (or implied equal when not set in the csv file). Specify the DA_Assoc parameter for the write Map Descriptor's, ensuring that non-associated Map Descriptor's have different values for this parameter.7
DAP:#26	Err. Map Descriptor= <%s> Read: param->daire_function= %c	You are trying to a read/write data from a DAP device and the driver cannot process the specified DA_Function for reading. 7
DAP:#27	Err. Cant write. function= %d Map Descriptor= <%s>	
DAP:#28!	Err. No Map Descriptor's	Report this error to FST providing a copy of your CSV file.
DAP:#29	Err. Map Descriptor= <%s> Special specified with invalid method.	Read Appendix B.4 and review the CSV file. When you have corrected the CSV file reset the FieldServer for the changes to take effect.
DAP:#30	Err. Map Descriptor= <%s> Unknown translation Method= %d	

⁷ Edit the CSV file and reset the FieldServer for the changes to take effect.

Message	Description	Discussion
DAP:#31	Err. Cant translate map desc. Data not stored	Make a log file and report the problem to FST providing a copy of your CSV file.
DAP:#32	Err. Driver does not support unit type='%c' (node= %d)	The driver found a device whose type is not supported. Remove this node from your CSV file and reset the FieldServer
DAP:#33	Err. Conflicting unit types='%c' vs '%c' (node= %d)	The expected and discovered unit type are different. The driver will always work with the discovered unit type.
DAP:#35	Err. Cant write to a data logger Map Descriptor= <%s>	Correct the problem by removing the offending map desc and reset the FieldServer for the change to take effect
DAP:#36	FYI. Can't write until unit type is known. Will retry. Map Descriptor= <%s>	This message is printed when you have specified a write with a DA_Field but no DA_Function. Under these circumstances the driver cannot write until it knows the Unit Type of the device. If you are polling using the 'Everything' function then the driver will obtain the unit type in time and the error will not be reproduced.
DAP:#37	Err. Map Descriptor= <%s> Cant write '%s' (%d:%d) to a unit= %d	The DA_Field and DA_Function type specified are not suitable for the Unit_Type found
DAP:#38	FYI. Configured/Reported Unit Types don't match. Node= <%s>	This message may be safely ignored. If the discovered type is what you expect then change the CSV file and reset the FieldServer for you changes to have effect.
DAP:#39	FYI. Poll Delay set to min of %.1f secs	You may safely ignore this message. If is for your information only. If you set the poll delay too short the driver will overwrite your value. The message reports the fact and indicates the minimum poll delay. You should note the poll delay as it affects performance.
DAP:#39b	FYI. Poll Delay override for DEBUG = %.1f secs	If you see this message printed report the fact to FST immediately.
DAP:#40	FYI. Diagnostic Poll Sent. Map Descriptor= <%s>	You may safely ignore this message. If is for your information only. It records the fact that a special poll has been sent.
DAP:#41	Err. Write abandoned. Map Descriptor=<%s> Addr=%d Only DAP80-Menu Write Thru's Supported	From 1.06d some write through capability was provided when variables in the 'Everything' array are updated. Only those variables which result in a DAP80-MENU command being sent are supported.
DAP:#42	Err. Write abandoned. Map Descriptor=<%s> Addr=%d. Map Entry not found	The Write Through Failed because the map entry could not be found. The map entry index is calculated by using the cache Map Descriptor normalized address and subtracting the Data Array offset. If the method is failing with error adjust your configuration so that the Data Array Offset is zero. After downloading the modified config and resetting the FieldServer if the error still occurs then take a log and send the log with the configuration to FieldServer's Tech support

Appendix D.1. Exposing Driver Statistics

In addition to the standard FieldServer communication statistics described in the FieldServer Configuration Manual, the DAP Driver can also expose some driver statistics by writing data to a Data Array. A special Map Descriptor named "dap-stats" is required.

Nodes			
Node_name	, Protocol	, Station	
null_node	, DAP	, 1	
Data_Arrays			
Data_Array_Name	, Data_Format,	, Data_Array_Length	
DA_DAP_STATS	, UINT32	, 1000	
Map_Descriptors			
Map_Descriptor_Name	, Data_Array_Name	, Node_name	, length
dap-stats	, DA_DAP_STATS	, null_node	, 1000

The driver uses the Data Array DA_DAP_STATS (in this example) to store driver specific statistics in the dap-stats Map Descriptor. Only one of these Map Descriptors may be specified per FieldServer.

The driver stores the following data

1	DRV_DLL_CLIENT_SENDS_MSG
2	DRV_DLL_CLIENT_SENDS_ACKNAK
3	DRV_DLL_CLIENT_SENDS_BYTES
4	DRV_DLL_SERVER_SENDS_MSG
5	DRV_DLL_SERVER_SENDS_ACKNAK
6	DRV_DLL_SERVER_SENDS_BYTES
7	DRV_DLL_CLIENT_RCVS_MSG
8	DRV_DLL_CLIENT_RCVS_BYTES
9	DRV_DLL_SERVER_RCVS_MSG
10	DRV_DLL_SERVER_RCVS_BYTES
11	DRV_DLL_TIMEOUT
12	DRV_DLL_ERROR
13	DRV_DLL_ERROR_CODE
14	DART_STAT_FAILED_CHECKSUM
15	DART_STAT_FAILED_FUNCTION
16	DART_STAT_FAILED_NOSTART
17	DART_STAT_FAILED_PROTOCOL
18	DART_STAT_FAILED_IC_TIMEOUT
19	DART_STAT_NOISE_BYTE_COUNT
20	DART_STAT_NOISE_BYTE_VALUE
21	DART_STAT_BUFFER_OVERFLOW
22	DART_SPECIAL_COMMAND_COUNT
23	DART_SPECIAL_COMMAND_CODE

24	DART_MSG_IGNORED_NO_SUITABLE_M
25	DART_FAILED_PARSE
26	DART_CANT_WRITE1
27	DART_CANT_WRITE2
28	DART_MSG_IGNORED_NOT_USEFUL
29	DART_STAT_FAILED_TIMEOUT1
30	DART_STAT_FAILED_TIMEOUT2
31	DART_IGNORE_MSG_TYPE
32	DART_TEST1
33	DART_TEST2
34	DART_MSG_UNEXPECTED_RESPONSE
35	DART_MSG_UNEXPECTED_CMAP
36	DART_MSG_UNEXPECTED_QUERY

Appendix E. WRITING DATA TO DAP DEVICES

Appendix E.1. Background

The variables in a DAP device are not individually addressable. When a DAP device is read a data composite is returned and the driver extracts the required data. When data is written to a DAP device it is not possible to write a value to one individual data element such as a temperature setpoint. Rather, the DAP devices requires the complete data composite (all its variables, states, settings) be written at once. This makes the setting of a setpoint a complex operation for the driver.

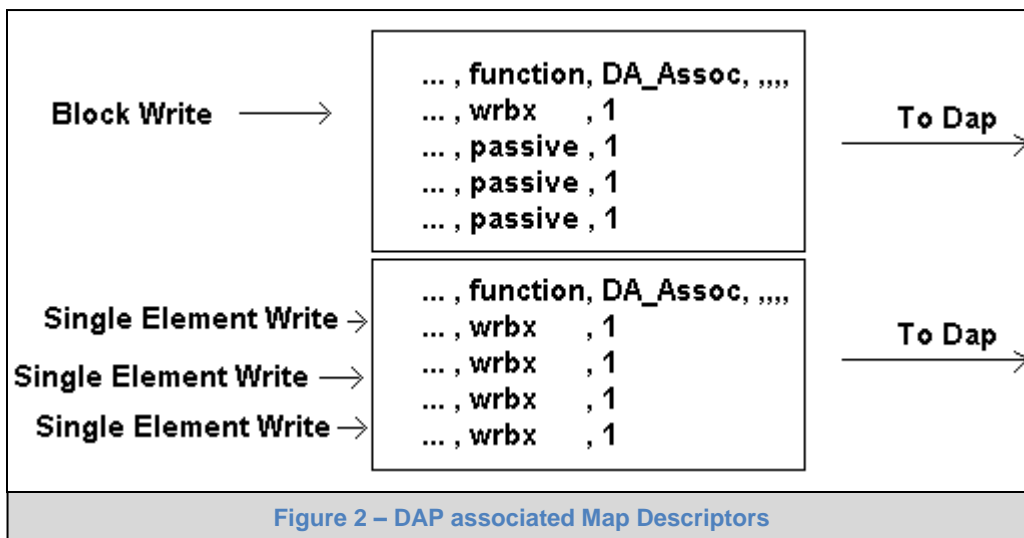
It involves the following steps

- a. Read the device, obtain a complete set of data and store (internally in the driver.).
- b. Use this stored data to form the basis for a write. Modify the data with the data the user wishes to set.
- c. Write the modified data composite back to the DAP device.

Thus to complete a 'write' operation successfully, the device must be read successfully. If the read has not been completed then the write operation will be abandoned. The driver prints messages to the error logs and records a NODE_OFFLINE stat under these circumstances.

Appendix E.2. Associated Map Descriptors

When writing to a DAP device it is possible to associate Map Descriptors using the DA_Assoc parameter. This is straightforward using WRBC or RDBC. When using WRBX to trigger the writes on change, however, the function of the associated Map Descriptors must be passive if the upstream device block writes the data. The following diagram outlines the required strategy.



Appendix E.3. Starting/Stopping DAP devices

The **DAP-Config** command is an **exception** to the above notes. A read is not required. The DAP Config command is used to turn off/on DAP units. The Command uses two consecutive array elements. The first is the zone, the second is the inhibit command. Valid zones are 0-63.

If any inhibit bit (bits 0-5) is set then the unit will not run. If bits 0 to 5 are off then the unit will run. Bits 6& 7 are used for display only on the DAP panel.

0x01 - Inhibit Cooling

0x02 - Inhibit Heating

0x04 - Inhibit Humid

0x08 - Inhibit Dehumidification

0x10 - Inhibit fan

0x20 - Reserved

0x40 - Network Standby - Display on panel (only has no effect on unit)

0x80 - Network off inhibit is active - display only - no effect on operation

This function should be used as a wrbc/x. When the driver encounters this command it reads the associated Data Array, loads the two elements found at the array offset into the message and transmits the message. The driver does not set any Data Array elements to confirm that the command concluded successfully.

Additional information is provided in Appendix B.10

Appendix E.4. DAP Unit Errors

Dap-Stat Query

The following table provides information on the storage of errors obtained in response to the Dap-Stat Query. Note that the errors are stored two ways – One storage method stores 8 alarm bits at a time in a single array element (stored as byte values) – The other method stores each alarm at a different location in the Data Array.

Byte	Bit	Relative Offset	Location in Data Array (Absolute Offset) Each alarm stored as a separate bit	Location in Data Array (Absolute Offset) Each set of 8 alarms stored as a byte*	046, 048 and 049 Units	DAP 80 Units
0	0	0	73	44:00	N/A	Manual Override: check bypass switch (DapII ver 5.0 or later)
0	1	1	74	44:01	High Pressure Failure Compressor No. 1	Water detected under floor: check probe
0	2	2	75	44:02	High Pressure Failure Compressor No. 2	No air flow: check belt and motor
0	3	3	76	44:03	High Pressure Failure Compressor No. 3	Dirty filter: check filters
0	4	4	77	43:00	High Pressure Failure Compressor No. 4	Humidifier problem: check water pressure
0	5	5	78	43:01	High Pressure Warning Compressor No. 1	Low voltage warning: check unit
0	6	6	79	43:02	High Pressure Warning Compressor No. 2	Firestat alarm: unit shutdown
0	7	7	80	43:03	High Pressure Warning Compressor No. 3	Compressor short cycle warning
1	0	8	81	46:00	High Pressure Warning Compressor No. 4	Power problem or unit restart
1	1	9	82	46:01	Low Pressure Failure Compressor No. 1	Humidity sensor problem
1	2	10	83	46:02	Low Pressure Failure Compressor No. 2	Temperature sensor problem
1	3	11	84	46:03	Low Pressure Failure Compressor No. 3	Scheduled maintenance due
1	4	12	85	44:04	Low Pressure Failure Compressor No. 4	High pressure C1: Manual reset required
1	5	13	86	44:05	No Air Flow, Check Belt and motor	Low pressure C1: Manual reset required
1	6	14	87	44:06	Change Filters, High Pressure Drop	High pressure C2: Manual reset required

Byte	Bit	Relative Offset	Location in Data Array (Absolute Offset) Each alarm stored as a separate bit	Location in Data Array (Absolute Offset) Each set of 8 alarms stored as a byte*	046, 048 and 049 Units	DAP 80 Units
1	7	15	88	44:07	Local Alarm. See Tag inside Door	Low pressure C2: Manual reset required
2	0	16	89	45:00	Manual Override, service or emergency	Smoke detector: unit shutdown
2	1	17	90	45:01	Humidifier Failure, Check water pressure	No water flow: check pump
2	2	18	91	45:02	No water flow. Check glycol pump.	Discharge temperature sensor problem
2	3	19	92	45:03	Smoke detected, located inside unit	High temperature warning
2	4	20	93	45:04	Fire Stat Tripped, Location inside unit	Low temperature warning
2	5	21	94	45:05	Standby pump on, check glycol pump	High humidity warning
2	6	22	95	45:06	Low voltage warning, check unit	Low humidity warning
2	7	23	96	45:07	Water detected in under floor area	Fan motor overload: check motor amperage
3	0	24	97	46:00	Low air temperature warning	Local alarm #1: See tag inside door
3	1	25	98	46:01	High air temperature warning	Local alarm #2: See tag inside door
3	2	26	99	46:02	Low Humidity warning	Local alarm #3: See tag inside door
3	3	27	100	46:03	High Humidity warning	Local alarm #4: See tag inside door
3	4	28	101	46:04	Low chilled-water-in alarm	Standby pump on: check primary pump
3	5	29	102	46:05	High chilled-water-in alarm	UPS power on: check main power supply
3	6	30	103	46:06	Low chilled-water-out alarm	Custom message on optional input #1
3	7	31	104	46:07	High chilled-water-out alarm	Custom message on optional input #2
4	0	32	105	47:00	High or Low limits misadjusted	Custom message on optional input #3
4	1	33	106	47:01	Air temperature sensor error	Custom message on optional input #4
4	2	34	107	47:02	Humidity sensor error	Humidification Inhibited
4	3	35	108	47:03	Discharge air sensor error	Reheat inhibited
4	4	36	109	47:04	No communications with temp/humid module	Reheat and humidification inhibited
4	5	37	110	47:05	No communications with LCD module	Discharge air temperature limit (just a condition)

Byte	Bit	Relative Offset	Location in Data Array (Absolute Offset) Each alarm stored as a separate bit	Location in Data Array (Absolute Offset) Each set of 8 alarms stored as a byte*	046, 048 and 049 Units	DAP 80 Units
4	6	38	111	47:06	Alarm network communications failure	Reheat mode during dehumidification (just a condition)
4	7	39	112	47:07	AC Power or controller restart	Manual Override: Check bypass switches (DAPII ver 5.1 and later)
5	0	40	113	48:00	Network doppelganger error	N/A
5	1	41	114	48:01	Incompatible Fahrenheit / centigrade mode	N/A
5	2	42	115	48:02	Schedule normal maintenance due	N/A
5	3	43	116	48:03	compressor short cycle warning	N/A
5	4	44	117	48:04	Fan motor overload, check motor amperage	N/A
5	5	45	118	48:05	Custom message on optional alarm input #1	N/A
5	6	46	119	48:06	Custom message on optional alarm input #2	N/A
5	7	47	120	48:07	Custom message on optional alarm input #3	N/A
6	0	48	121	49:00	Discharge air temperature limit (just a condition)	N/A
6	1	49	122	49:01	Reheat mode during dehumidification (just a condition)	N/A
6	2	50	123	49:02	N/A	N/A
6	3	51	124	49:03	N/A	N/A
6	4	52	125	49:04	N/A	N/A
6	5	53	126	49:05	N/A	N/A
6	6	54	127	49:06	N/A	N/A
6	7	55	128	49:07	N/A	N/A
				Notes		
				*	43:00 Indicates array element zero.	

Byte	Bit	Relative Offset	Location in Data Array (Absolute Offset) Each alarm stored as a separate bit	Location in Data Array (Absolute Offset) Each set of 8 alarms stored as a byte*	046, 048 and 049 Units	DAP 80 Units
					Individual alarm can be found by masking and looking at the zero'th bit of this element.	

Chiller Stat Query

Unlike the Dap-Stat errors, these are stored in the Data Array using only one method – each error is stored at a separate location in the Data Array.

Location in Data Array (Absolute Offset) Each alarm stored as a separate bit	Alarm / Bit Descriptions	Byte	Bit	Relative Offset
536	Auxiliary cooling available	0	0	0
537	Auxiliary cooling locked out	0	1	1
538	Unit on backup module standby	0	2	2
539	Custom message on optional input #1	0	3	3
540	Custom message on optional input #2	0	4	4
541	Evaporator freeze stat module #1	0	5	5
542	Evaporator freeze stat module #2	0	6	6
543	Evaporator freeze stat module #3	0	7	7
544	High pressure problem module #1	1	0	8
545	High pressure problem module #2	1	1	9
546	High pressure problem module #3	1	2	10
547	High return temperature warning	1	3	11
548	High supply temperature warning	1	4	12
549	Local alarm #1: See tag inside door	1	5	13
550	Local alarm #2: See tag inside door	1	6	14
551	Low pressure problem module #1	1	7	15
552	Low pressure problem module #2	2	0	16
553	Low pressure problem module #3	2	1	17
554	Low return temperature warning	2	2	18

Location in Data Array (Absolute Offset) Each alarm stored as a separate bit	Alarm / Bit Descriptions	Byte	Bit	Relative Offset
555	Low supply temperature warning	2	3	19
556	Low voltage warning: check unit	2	4	20
557	Scheduled normal maintenance due	2	5	21
558	Manual override: compressor	2	6	22
559	Manual override: check bypass switches	2	7	23
560	No chilled water flow module #1	3	0	24
561	No chilled water flow module #2	3	1	25
562	No chilled water flow module #3	3	2	26
563	No condenser water flow	3	3	27
564	Power problem or restart	3	4	28
565	Return temperature sensor problem	3	5	29
566	compressor short cycle warning	3	6	30
567	Standby pump on: Check primary pump	3	7	31
568	Supply temperature sensor problem	4	0	32
569	Unit on total standby	4	1	33
570	Under floor water detected	4	2	34
571	N/A	4	3	35
572	N/A	4	4	36
573	N/A	4	5	37
574	N/A	4	6	38
575	N/A	4	7	39
576	N/A	5	0	40
577	N/A	5	1	41
578	N/A	5	2	42
579	N/A	5	3	43
580	N/A	5	4	44
581	N/A	5	5	45
582	N/A	5	6	46
583	N/A	5	7	47

Appendix E.5. Using DA_Assoc correctly with Writes/Toggling Values

Association of Map Descriptors can result in zero's being written to the DAP device. The following information can help to prevent this problem. When Map Descriptors are associated using the same value for the DA_Assoc parameter then the associated Map Descriptors are all processed before a write is sent to the DAP unit. A write to a DAP unit writes a complex set of data – it is not possible to write a single variable's value to a DAP device. Each associated Map Descriptor is used to update the complex set of data that will be written.

Consider the following Map Descriptor fragments

Data_Array	Data_Array_Offset	Function	Node_Name	Address	DA_Field	DA_Assoc	Scan_Interval
DA_SETPOINTS	0	WRBC	Unit1	0	1	nom_h_setpt	120s
DA_SETPOINTS	1	passive	Unit1	0	1	t_setpt	120s

If an upstream device writes a value to DA_SETPOINTS Offset 0, when the write is done the values of DA_SETPOINTS Offset 0 AND Offset 1 are both used to build the complex data set since the Map Descriptors are associated. If the upstream device has never set the value of DA_SETPOINTS offset 1, the value will be zero and hence a value of zero would be written to the DAP's Temperature Setpoint.

This could be avoided by not associating the Map Descriptors. This will, however, result in extra write messages being sent to the DAP device. The extra messages could be avoided by changing the wrbc to a wrbx. If you do this you must still consider the effects of association and in this case it may be best to have each Map Descriptor defined as a wrbx with a different value for DA_Assoc.

Appendix F. WRITE THROUGH'S

This driver provides very limited Write Through Support.

It is possible to generate a Write Through by writing to the Data Array used in a read Map Descriptor where the DA_Func keyword is 'Everything' provided that the write through results in Dap80-Menu command being sent to the DAP device. This is true for variables whose offsets are between 899 and 1086 in Section Appendix A.3