



---

A Sierra Monitor Company

**Driver Manual**  
**(Supplement to the FieldServer Instruction Manual)**

**FS-8700-65 Silent Knight**

**APPLICABILITY & EFFECTIVITY**

**Effective for all systems manufactured after January 1, 1999**

<b>Driver Version:</b>	<b>1.00</b>
<b>Document Revision:</b>	<b>7</b>

## TABLE OF CONTENTS

<b>1.</b>	<b>Silent Knight Gateway Command (SKgwcmd) Driver Description</b> .....	<b>4</b>
<b>2.</b>	<b>Driver Scope of Supply</b> .....	<b>5</b>
2.1.	Supplied by FieldServer Technologies for this driver .....	5
<b>3.</b>	<b>Hardware Connections</b> .....	<b>6</b>
<b>4.</b>	<b>Configuring the FieldServer as a Silent Knight Gateway Command (SKgwcmd) Driver Client</b> .....	<b>8</b>
4.1.	Data Arrays.....	8
4.2.	Client Side Connection Descriptions .....	9
4.3.	Client Side Node Descriptors .....	9
4.4.	Client Side Map Descriptors .....	10
4.4.1.	<i>FieldServer Specific Map Descriptor Parameters</i> .....	10
4.4.2.	<i>Driver Specific Map Descriptor Parameters</i> .....	10
4.4.3.	<i>Timing Parameters</i> .....	12
4.4.4.	<i>Map Descriptor Example 1 – General System Status Query.</i> .....	13
4.4.5.	<i>Map Descriptor Example 2 – Alarm Silence Command</i> .....	14
4.4.6.	<i>Map Descriptor Example 3 – Point Descriptions</i> .....	14
4.4.7.	<i>Map Descriptor Example 4 – Point Status</i> .....	15
4.4.8.	<i>Map Descriptor Example 5 – Point Enable/Disable</i> .....	16
<b>5.</b>	<b>Configuring the FieldServer as a Silent Knight Gateway Command (SKgwcmd) Driver Server</b> .....	<b>17</b>
5.1.	Data Arrays.....	17
5.2.	Server Side Connection Descriptors.....	18
5.3.	Server Side Node Descriptors .....	19
5.4.	Server Side Map Descriptors.....	20
5.4.1.	<i>FieldServer Specific Map Descriptor Parameters</i> .....	20
5.4.2.	<i>Driver Specific Map Descriptor Parameters</i> .....	20
5.4.3.	<i>Timing Parameters</i> .....	21
5.4.4.	<i>Map Descriptor Example – FieldServer responds to Point Status Requests</i> .....	22
<b>Appendix A.</b>	<b>.....</b>	<b>23</b>
Appendix A.1.	Legal Values for Sk_function.....	23
Appendix A.2.	Length Parameter Settings based on sk_function parameter .....	23
<b>Appendix B. Protocol Commands</b> .....	<b>.....</b>	<b>24</b>
Appendix B.1.	General System Status .....	24
Appendix B.2.	System Trouble Status .....	24
Appendix B.3.	General Zone Status .....	25
Appendix B.4.	Zone Description .....	25
Appendix B.5.	Output Group Status .....	25
Appendix B.6.	Output Group Description.....	26
Appendix B.7.	Point Status .....	26
Appendix B.8.	Point Description .....	27
Appendix B.9.	Time Status .....	27
Appendix B.10.	Gateway Version .....	27
Appendix B.11.	Disable Point / Enable Point.....	27
Appendix B.12.	Silence.....	27
Appendix B.13.	Reset Alarms.....	28
Appendix B.14.	Time Synch .....	28

**Appendix C. Driver Error Messages ..... 29**  
Appendix C.1. Driver Stats ..... 31

## 1. Silent Knight Gateway Command (SKgwcmd) Driver Description

The Silent Knight Gateway Command (SKgwcmd) Driver allows the FieldServer to transfer data to and from devices over either RS-232 or RS-485 using Silent Knight Gateway Command (SKgwcmd) Driver protocol. The FieldServer can emulate either a Server or Client.

The Silent Knight Gateway Command (SKgwcmd) Driver is based on the p[rotocol specification "Gateway Command Set Specification 5820 Intellinight System rev D.2 dated 11 July 2000"

The serial port on the Silent Knight Unit's is not addressable and the protocol does not support node identification. Therefore, only one Silent Knight Unit can be connected to each port on the FieldServer.

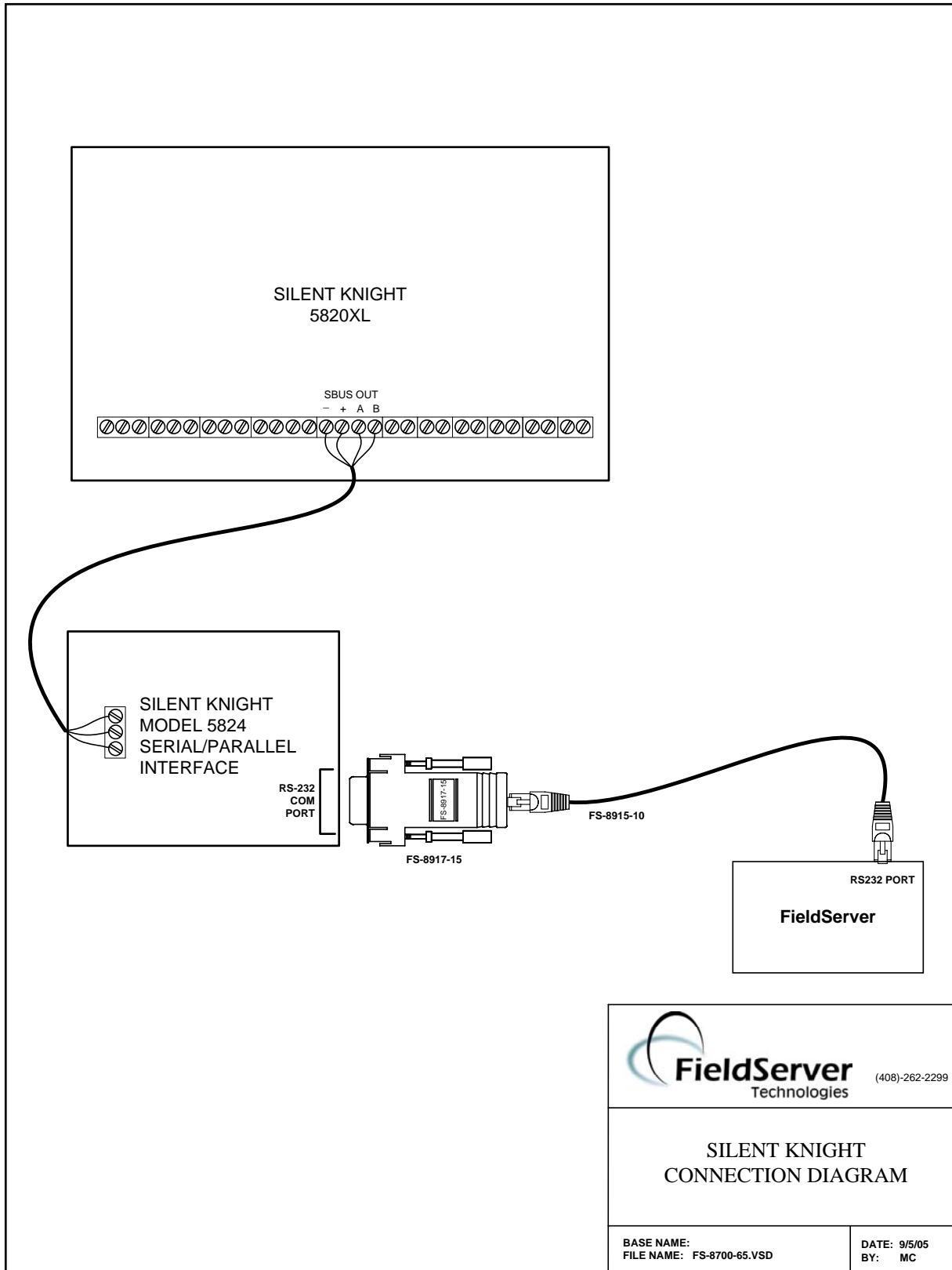
## 2. Driver Scope of Supply

### 2.1. Supplied by FieldServer Technologies for this driver

FieldServer Technologies PART #	Description
FS-8917-15	RJ45 to DB9F connector adapter
SPA59132	RS-485 connection adapter
FS-8700-65	Driver Manual.

### 3. Hardware Connections

The FieldServer is connected to the Silent Knight Gateway Command Driver as shown below. Configure the Silent Knight Gateway Command Driver according to manufacturer's instructions



#### 4. Configuring the FieldServer as a Silent Knight Gateway Command (SKgwcmd) Driver Client

For a detailed discussion on FieldServer configuration, please refer to the FieldServer Configuration Manual. The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer (See “.csv” sample files provided with the FieldServer).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a Silent Knight Gateway Command (SKgwcmd) Driver Server.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for Silent Knight Gateway Command (SKgwcmd) 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.

##### 4.1. Data Arrays

Section Title		
Data_Arrays		
Column Title	Function	Legal Values
Data_Array_Name	Provide name for Data Array	Up to 15 alphanumeric characters
Data_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 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

Section Title		
Connections		
Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2 <sup>1</sup> .
Baud*	Specify baud rate	110 – 115200, standard baud rates only
Parity*	Specify parity	Even, Odd, <b>None</b> , Mark, Space
Data_Bits*	Specify data bits	7, <b>8</b>
Stop_Bits*	Specify stop bits	<b>1</b>
Protocol	Specify protocol used	SKgwcmd
Handshaking*	Specify hardware handshaking	RTS, RTS/CTS, <b>None</b>
Poll Delay*	Time between internal polls	0-32000 seconds, <b>1s</b>

#### Example

// Client Side Connections			
Connections			
Port,	Baud ,	Parity,	Protocol
P8,	19200,	None,	SKgwcmd

### 4.3. Client Side Node Descriptors

Section Title		
Nodes		
Column Title	Function	Legal Values
Node_Name	Provide name for node	Up to 32 alphanumeric characters
Node_ID*	This commonly used parameter is not required or used by this driver.	
Protocol	Specify protocol used	SKgwcmd
Port	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2 <sup>1</sup>

#### Example

// Client Side Nodes		
Nodes		
Node_Name,	Protocol,	Port
SK-Device1,	SKgwcmd,	P8

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

#### 4.4. Client Side Map Descriptors

##### 4.4.1. FieldServer 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	Data Array name from section 4.1
Data_Array_Location	Starting location in Data Array	0 to maximum specified in section 4.1.
Function	Function of Client Map Descriptor When using the alarm silence and reset commands the use of WRBX is recommended. Additional notes are provided in section 1.	RDBC, WRBC, WRBX
Da_byte_name*	Only used when sk_function is one of the following: <i>Output Group Status</i> <i>Point Status</i> <i>Point Description</i> See sections Appendix B.5 to Appendix B.7 for additional information.	One of the Data Array names from section 4.1.
DA_bit_name*	Only used when sk_function is <i>Point Description</i> . See Appendix B.7	Data Array name from section 4.1

##### 4.4.2. Driver Specific Map Descriptor Parameters

Column Title	Function	Legal Values
Node_Name	Name of Node to fetch data from	One of the node names specified in Section 4.3.
Data_Type	Data type	Register, Coil, AI, DI
Length	Length of Map Descriptor	1 – 1000 The length must be sufficient to store all data polled to prevent data being discarded by the driver. Certain commands, e.g. description queries use 20 array elements to store the response. Thus, the length must be at least 20.
Address*	This commonly used parameter is not required by this driver.	
Sk_function	This mandatory field tells the driver what query or command is required..	Refer to Appendix A.1

Column Title	Function	Legal Values
Sk_zone*	<p>This parameter is required for <i>General Zone Status</i> <i>Expanded Zone Status</i> <i>Zone Description</i> Specify the number of the zone being queried. Refer to Appendix B.3 and Appendix B.4</p>	<p>Whole numbers limited to the range of addressable zone configured in the Silent Knight device.</p>
Sk_group*	<p>This parameter is required for the following commands.  <i>Output Group Status</i> <i>Output Group Description</i> Refer to Appendix B.5 and Appendix B.6.</p>	
Sk_module*	<p>This parameter is always used in conjunction with the sk_point parameter to uniquely define the address of a point to be queried, enabled or disabled.  Additional Notes are provided in Appendix B.7 and Appendix B.8</p>	<p>0, 1 ,2 ...  Whole numbers limited to the addressable expander Module ID's.</p>
Sk_point*	<p>When querying a specific point both the sk_module and sk_point parameters must be defined.  When using the <i>Point Description</i> function a single non-zero value must be specified and the sk_module parameter must be set to a non-zero value. This is because this function only queries one specific point per Map Descriptor.</p>	<p>0, 1 , 2 ... 250  Whole numbers limited to the range of addressable points in the Silent Knight Unit.</p>

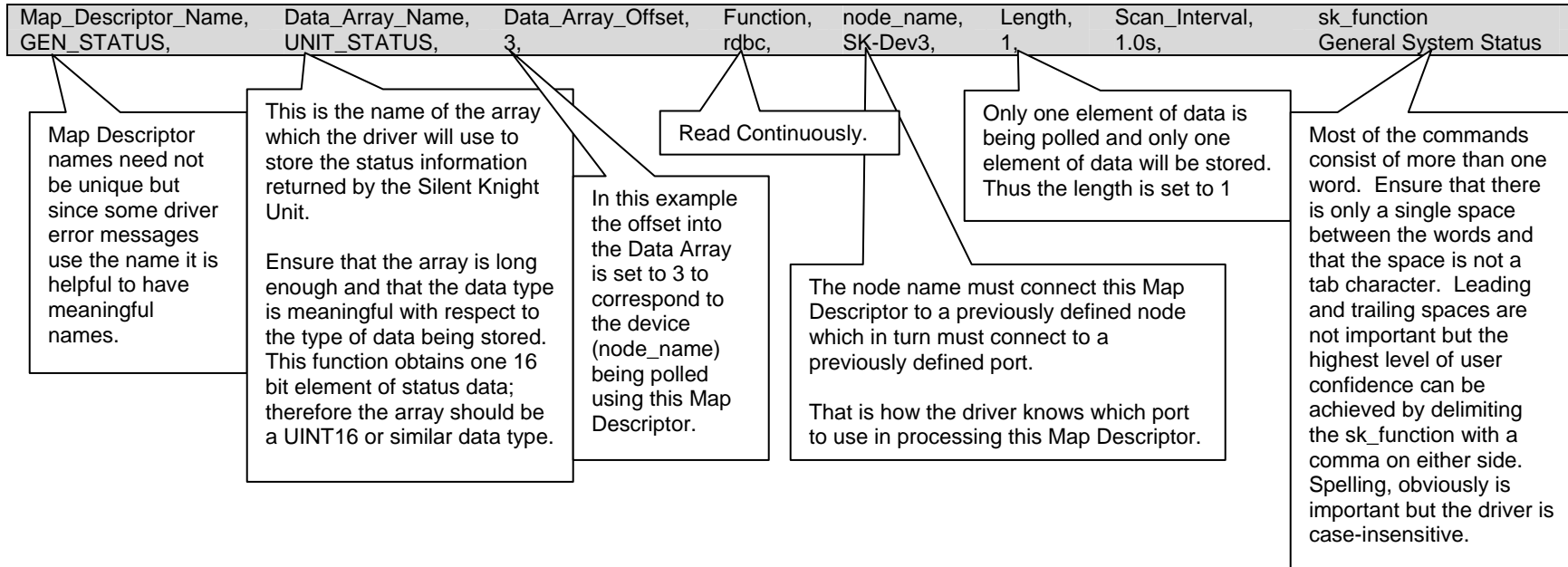
Column Title	Function	Legal Values
	<p>When using the <i>Point Status</i> function several possibilities arise.</p> <p>Set <i>sk_module</i> &amp; <i>sk_point</i> to zero The poll will query all possible points in the Silent Knight unit. Multiple Map Descriptors may be required to handle all the data that is returned.</p> <p><i>Set sk_module to a specific number &amp; set sk_point to zero</i> The unit will return data for all the points on the specified module. <i>Set sk_module &amp; sk_point to non-zero values and length to 1.</i> Driver will poll the status of one single point. Additional notes are provided in section 1.</p>	

**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 1 – General System Status Query.

The Map Descriptor causes the driver to poll the Silent Knight Device every 1.0 seconds, read the general system status and store the result in an array called UNIT\_STATUS



### 4.4.5. Map Descriptor Example 2 – Alarm Silence Command

This example illustrates the use of the silence command. The alarm reset command is used in an identical fashion. Use the WRBX function to trigger the command whenever a silence is required. This function causes the driver to watch the value of the Data Array specified, when the value changes the driver will perform the command. Typically, users configure the FieldServer to connect a MMI or field push button on a remote device to set the array element to 1 to trigger the silence command and then set it back to zero.

Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	Function,	node_name,	Length,	Sk_function
SILENCE-CMD,	CMD_TRIGGERS,	0,	wrbx,	SK-DEV1,	2,	Silence

Have some remote device set the value of this Data Array to trigger this command.

Write on change. When the value of the 1<sup>st</sup> element of the array called CMD\_TRIGGERS changes then the command will be triggered.

Use a length of 2 as a later release of the driver may write the return code from the SK device to the second element of the Data Array.

The Alarm Silence command will be sent to the SK device.

### 4.4.6. Map Descriptor Example 3 – Point Descriptions

The Map Descriptor below illustrates a query for a point description. Each query returns three data components – a point type, accessory type and a point description. Only one point can be queried per Map Descriptor.

Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	Function,	node_name,	Length,	Scan_Interval,	sk_function,	sk_module,	sk_point,	da_byte_name,	da_bit_name
PNT_DESC1,	DESC_ARRAY,	0,	rdbc,	SK-DEV1,	20,	1s,	Point Description ,	1,	5,	TYPE_ARR,	ACC_ARR

The point's description will be placed in 20 consecutive elements of this Data Array starting at the offset position.

Module = 1 point = 5. Both parameters must be non-zero.

The additional data components are stored in these Data Arrays.

Thus is how you tell the driver that 20 elements of the Data Array are controlled by this Map Descriptor.

Function name-spelling and spacing are important.

Only one point is processed for each of Point Description Map Descriptor.

The point type is stored as a number at the same offset (specified by Data\_array\_offset) in the array called TYPE\_ARR.

The accessory type is stored at the same offset in the array called ACC\_ARR.

### 4.4.7. Map Descriptor Example 4 – Point Status

This example reads all the point status data from the SK device. One active Map Descriptor polls the device. The remaining passive Map Descriptors are used for storage. The ‘length’ of each passive Map Descriptor has been set to a large number to accommodate all possible points.

Map Descriptors									
Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	Function,	node_name,	Length,	Scan_Interval,	sk_function,	sk_module,	sk_point
POLL,	DUMMY,	0,	rdbc,	SK-DEV1,	1,	5.0s,	Point Status,0,	0	

Map Descriptors										
Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	Function,	node_name,	Length,	Scan_Interval,	sk_function,	sk_module,	sk_point,	da_byte_name
MOD1,	DA_AI1,	0,	passive,	SK-DEV1,	200,	5.0s,	Point Status,	1,	0,	EXTRA_DATA1
MOD2,	DA_AI2,	0,	passive,	SK-DEV1,	200,	5.0s,	Point Status,	2,	0,	EXTRA_DATA2
MOD3,	DA_AI3,	0,	passive,	SK-DEV1,	200,	5.0s,	Point Status,	3,	0,	EXTRA_DATA3
MOD4,	DA_AI4,	0,	passive,	SK-DEV1,	200,	5.0s,	Point Status,	4,	0,	EXTRA_DATA4
MOD5,	DA_AI5,	0,	passive,	SK-DEV1,	200,	5.0s,	Point Status,	5,	0,	EXTRA_DATA5

This Map Descriptor generates the point status poll to be sent to the SK device.

The Status data is stored in these arrays. The offset is based on the point number and the offset defined in the Map Descriptor.

These Map Descriptors wait passively for returning data. They tell the driver how to store the data.

Length is set to 200 thus up to 200 points may have their data stored – if they exist. If point 201 was added to the system it would have its data discarded.

Require one Map Descriptor for each module that may return point data.

The accessory data associated with each point is stored in the corresponding location of these Data Arrays.  
  
If you do not specify this parameter then the accessory data is discarded.

### 4.4.8. Map Descriptor Example 5 – Point Enable/Disable.

In this example point 10 of module 1 is disabled. The command is executed each time the value of the array DA01 element index 10 (11<sup>th</sup> element) changes. If you want this command to be executed continuously then you could change the function to a wrbc. In a later version of this driver, the same data element will be set to 1 by the driver if the command did not complete successfully or it will be set to zero if the command completed successfully.

Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	Function,	node_name,	Length,	Scan_Interval,	sk_function,	sk_module,	sk_point
ENABLE1,	DA01,	10,	wrbx,	SK-DEV1,	1,	1.0s,	Point Disable,	1,	10

This is a triggered command.  
 The Data Array offset = 10.. Thus the element of the Data Array indexed by 10 (11<sup>th</sup> element) will trigger the command when it changes.

Change this to *Point Enable* to enable a point. The remainder of the Map Descriptor remains the same

Sk\_module and sk\_point must be non-zero. Only one point can be enabled or disabled per Map Descriptor.

## 5. Configuring the FieldServer as a Silent Knight Gateway Command (SKgwcmd) Driver Server

The Silent Knight Gateway Command (SKgwcmd) Driver can act as a Server. As the protocol does not support device or node identification it is not possible for this driver to handle more than one Client per FieldServer unless the driver is specifically tied to a port by using the port keyword when defining the Server side node.

For a detailed discussion on FieldServer configuration, please refer to the FieldServer Configuration Manual. The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer (See “.csv” sample files provided with the FieldServer).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a Silent Knight Gateway Command (SKgwcmd) Driver Client

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for Silent Knight Gateway Command (SKgwcmd) Driver communications, the driver independent FieldServer buffers need to be declared in the “Data Arrays” section, the FieldServer virtual node(s) needs to be declared in the “Server Side Nodes” section, and the data to be provided to the Clients needs to be mapped in the “Server Side Map Descriptors” section. Details on how to do this can be found below.

Note that in the tables, \* indicates an optional parameter, with the **bold** legal value being the default.

### 5.1. Data Arrays

Section Title		
Data_Arrays		
Column Title	Function	Legal Values
Data_Array_Name	Provide name for Data Array	Up to 15 alphanumeric characters
Data_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 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
```

**5.2. Server Side Connection Descriptors**

Section Title		
Connections		
Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2 <sup>2</sup> .
Baud*	Specify baud rate	110 – 115200, standard baud rates only
Parity*	Specify parity	Even, Odd, <b>None</b> , Mark, Space
Data_Bits*	Specify data bits	7, <b>8</b>
Stop_Bits*	Specify stop bits	<b>1</b>
Protocol	Specify protocol used	SKgwcmd
Handshaking*	Specify hardware handshaking	RTS, RTS/CTS, <b>None</b>

**Example**

```
// Server Side Connections
Connections
Port,                  Baud,              Protocol
P8,                   19200,             SKgwcmd
```

<sup>2</sup> Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

**5.3. Server Side Node Descriptors**

Section Title		
Nodes		
Column Title	Function	Legal Values
Node_Name	Provide name for node	Up to 32 alphanumeric characters
Node_ID	This commonly used parameter is not required by this driver.	
Protocol	Specify protocol used	SKgwcmd

**Examples**

```
// Server Side Nodes

Nodes
Node_Name,          Protocol
FieldServer,        SKgwcmd
```

```
// Use this example when the FieldServer is being used to
// emulate more than one Silent Knight Device.

Nodes
Node_Name,          Protocol,          Port
Sk-emulation1,     SKgwcmd,          P1
Sk-emulation2,     SKgwcmd,          P2
```

## 5.4. Server Side Map Descriptors

### 5.4.1. 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_Location	Starting location in Data Array	0 to maximum specified in "Data Array" section above
Function	Function of Client Map Descriptor	PASSIVE
Da_byte_name*	Only used when sk_function is one of the following: <i>Output Group Status</i> <i>Point Status</i> <i>Point Description</i>  See Appendix B.5 to Appendix B.7 for additional information.	One of the Data Array names from "Data Array" section above
DA_bit_name*	Only used when sk_function is <i>Point Description</i> .  See Appendix B.7 for additional information.	One of the Data Array names from "Data Array" section above

### 5.4.2. Driver Specific Map Descriptor Parameters

Column Title	Function	Legal Values
Node_Name	Name of Node to fetch data from	One of the node names specified in "Client Node Descriptor" above
Data_Type	Data type	Register, Coil, AI, DI
Length	Length of Map Descriptor  The length parameter's meaning is dependent on the value of the sk_function parameter. Refer to Appendix A.2	1 - 1000
Address	This parameter is not required by this driver.	
Sk_function		Refer to Appendix A.1
Sk_zone*	Use one Map Descriptor for each zone to be emulated.	1, 2, 3 ....
Sk_group*		Positive Integers
Sk_module*	These parameters are linked as points belong to modules. Thus the point and module define each point uniquely.	1, 2, 3 ....
Sk_point*		Positive Integers

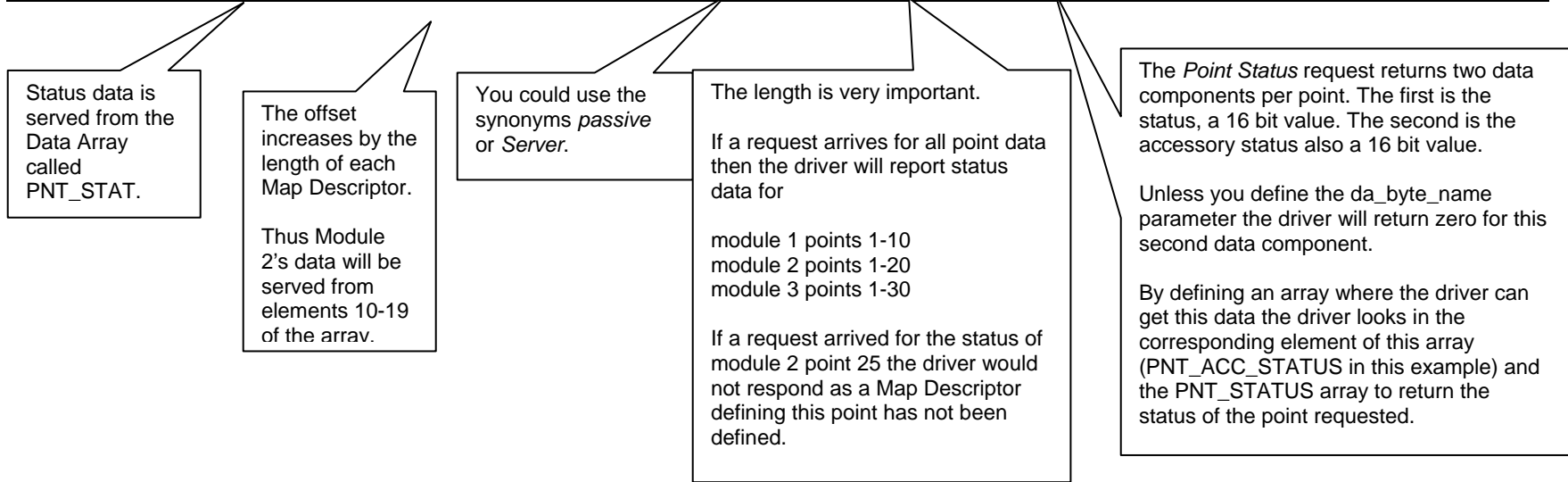
### 5.4.3. Timing Parameters

Column Title	Function	Legal Values
Scada_Hold_Timeout	Time Server side waits before responding to Client that node is offline on FieldServer Client side.	>1.0s

### 5.4.4. Map Descriptor Example – FieldServer responds to Point Status Requests

This example illustrates a number of Map Descriptors used to allow the FieldServer to emulate 3 modules of 10 points each. These Map Descriptors can only respond to **Point Status** requests. Other Map Descriptors are required to allow responses to other queries like **point description** queries.

Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	Function,	node_name,	Length,	sk_function,	sk_module,	sk_point,	da_byte_name
SERVER1,	PNT_STAT,	0,	passive,	Node_A,	10,	Point Status,	1,	1,	PNT_ACC_STATUS
SERVER2,	PNT_STAT,	10,	passive,	Node_A,	20,	Point Status,	2,	1,	PNT_ACC_STATUS
SERVER3,	PNT_STAT,	30,	passive,	Node_A,	10,	Point Status,	3,	1,	PNT_ACC_STATUS



---

**Appendix A.**


---

**Appendix A.1. Legal Values for Sk\_function**

General System Status  
 System Trouble Status  
 General Zone Status  
 Expanded Zone Status  
 Zone Description  
 Output Group Status  
 Output Group Description  
 Point Status  
 Point Description  
 Time Status  
 Gateway Version  
 Disable Point  
 Enable Point  
 Reset Alarms  
 Silence  
 Time Synch

**Appendix A.2. Length Parameter Settings based on sk\_function parameter**

<b>Sk_function command</b>	<b>Length Parameter Meaning</b>
<i>Silence</i> <i>Reset Alarms</i>	The length parameter tells the driver how much of the Data Array to zeroize when one of these commands is received
<i>Output Group Status</i> <i>General Zone Status</i>	The length parameter tells the driver how many zone or groups to return data for (if all zones or groups are requested) when one of these commands is received.
<i>Point Status</i>	The length tells the driver how many points to report if the poll requests data for all points.

For all other values of sk\_function the length should be set to 1 as the driver does not use the parameter.

## Appendix B. Protocol Commands

The following list of commands has been implemented in the Silent Knight Gateway Command (SKgwcmd) Driver. The table contains the keywords that may be used in the .CSV files to specify the sk\_function parameter.

	Commands Implemented.		Commands Implemented
1	General System Status	9	Point Description
2	System Trouble Status	10	Time Status
3	General Zone Status	11	gateway Version
4	Expanded Zone Status	12	Disable Point
5	Zone Description	13	Enable Point
6	Output Group Status	14	Reset Alarms
7	Output Group Description	15	Silence
8	Point Status	16	Time Synch

### Appendix B.1. General System Status

Requests the status of the entire system - the bits provided in the response will exactly mimic the state of the five indicator LED's on the IntelliKnight front panel.

Bit	Meaning	Bit	Meaning
0	Alarm condition present	8	Not Used
1	Supervisory condition present	9	Not Used
2	Trouble condition present	10	Not Used
3	System silenced	11	Not Used
4	AC Power present	12	Not Used
5	Not Used	13	Not Used
6	Not Used	14	Not Used
7	Not Used	15	Not Used

### Appendix B.2. System Trouble Status

Monitors the system for trouble or service requirements - the panel responds with the system trouble status record.

Bit	Meaning	Bit	Meaning
0	AC Trouble	8	Printer Trouble
1	Battery Trouble	9	Auxiliary Power Trouble
2	Earth Ground Fault Trouble	10	System Aux 1 Switch Trouble
3	unused	11	System Aux 2 Switch Trouble
4	Phone Line 1 Trouble	12	System Switch Trouble
5	Phone Line 2 Trouble	13	Undefined SLC Point Trouble
6	IO Device Trouble	14	SBUS Class A Trouble
7	Account Trouble	15	unused

### Appendix B.3. General Zone Status

Requests the status of one or more zones in the system - the panel responds with a zone status byte for each of the zone(s) specified.

- Set zone number to zero and length to 1 to request all possible zones
- Set zone number to a specific zone and length to 1 to read status data from one zone.
- Set zone number to a specific zone and length greater than one to read consecutive zones starting at the zone number specified.

Bit	Meaning	Bit	Meaning
0	Alarm condition present in the zone	8	Not Used
1	Supervisory condition present in the zone	9	Not Used
2	Trouble condition present in the zone	10	Not Used
3	Not Used	11	Not Used
4	Not Used	12	Not Used
5	Not Used	13	Not Used
6	Not Used	14	Not Used
7	Not Used	15	Zone Not Defined

### Appendix B.4. Zone Description<sup>3</sup>

Requests the text description & type designator for any individual zone in the system - the panel returns the zone description record for the zone specified. Only one specific zone may be queried. 40 consecutive array elements are loaded with the description returned by the device.

### Appendix B.5. Output Group Status

Requests information about the state of one or more output groups present in the system. Returns the active/inactive state and the number of troubles for each group.

- Set group number to zero and length to 1 to request all possible groups.
- Set group number to a specific group and length to 1 to read status data from one group.
- Set group number to a specific group and the length greater than one to read consecutive groups starting at the group number specified.

Value	Meaning
1	Group is Active
0	Group is Inactive
8000h	Group is not defined.

When an output group status is requested the SK device returns the status and the number of troubles in the group. The 'number of troubles' is discarded by this driver unless a second Data Array is provided to store this quantity. Specify the second Data Array by putting the array name as the value of a field called da\_byte\_name in the Map Descriptor. (See the usage for point status illustrated in Section 4.4.7.).

<sup>3</sup> As of 5820 release 6.0 (Gateway version 2.00), all text descriptions for zones are 16 bit Unicode instead of 8 bit ASCII. The number of bytes returned has been modified from gateway version 1.00

## Appendix B.6. Output Group Description

Requests the text description & type designator for any individual output group in the system - the panel returns the output group description record for the output group specified. See section 4.4.6 for additional information.

## Appendix B.7. Point Status

Requests the state of one or all points in the system - the panel returns the point status for the point(s) specified. This enables the interfacing computer to select for appropriate data from the large number of points possible in the IntelliKnight system.

### Point Status Values

Value	Meaning
0x00	Normal
0x01	Alarm
0x02	Pre-Alarm
0x03	Supervisory
0x04	Trouble
0x05	Output Active
0x06	Output Inactive
0x07	Output Disconnected
0x8000	Unused
0x8001	Undefined

### Accessory Status Values

Value	Meaning
0x00	Accessory Normal
0x01	Accessory Trouble
0x8000	Accessory Unused

- Set module number to zero to request point data for all points on all modules.
- Set module number non-zero and point number to zero to request data for all points on the specified module.
- Set module and point number non-zero to request data for a specific point. In this case only one point is requested.

When point status is requested the Silent Knight returns data elements per point. The first, (status), is stored using the name of the Data Array specified in the Map Descriptor. The second, (accessory status), is discarded unless the name of a second Data Array is specified in the *da\_byte\_name* field of the Map Descriptor. (See Section 4.4.7)

### Appendix B.8. Point Description<sup>4</sup>

Requests the text description & type designator of any individual point in the system - the panel returns the point description record for the point specified. One Map Descriptor is required for each point description requested.

When a point description is polled the returned data contains the point type, the accessory type and the description. The driver loads the description into 40 consecutive elements of the Map Descriptor's Data Array. The point type data is only stored if the parameter *da\_byte\_name* is specified by providing the name of another Data Array in which the point type data should be stored. The accessory type data is only stored if the parameter *da\_bit\_name* is specified by providing the name of another Data Array in which the accessory type data should be stored.

### Appendix B.9. Time Status

The time and date are stored in 6 consecutive element of the Data Array specified.

Element	Contents
1	Second
2	Minute
3	Hour
4	Day
5	Month
6	Year - 1900

### Appendix B.10. Gateway Version

Requests the gateway version number string - the panel responds with gateway version in a null terminated string. This command can be used to determine the command set supported by the connected panel.

The driver writes the version character by character to consecutive elements of the Data Array specified. The version is variable length so ensure that the Map Descriptor length is sufficient to cover all contingencies.

### Appendix B.11. Disable Point / Enable Point

A specific point must be specified by setting *sk\_module* and *sk\_point* to non-zero values.

### Appendix B.12. Silence

Used to silence any audible alarms and/or troubles occurring in the system -the panel returns a non-zero error code if the command could not be completed.

<sup>4</sup> As of 5820 release 6.0 (Gateway version 2.00), all text descriptions for points are 16 bit Unicode instead of 8 bit ASCII. The number of bytes returned has been modified from gateway version 1.00. Also, 5820 release 6.0 introduced a line of accessory bases. The accessory base type is now returned as part of this command

### Appendix B.13. Reset Alarms

Used to reset any alarms occurring in the system - the panel returns a non-zero error code if the command could not be completed.

### Appendix B.14. Time Synch

Used to set the time and date at the panel - the driver uses 6 consecutive elements from the Data Array specified.

Element	Contents
1	Second
2	Minute
3	Hour
4	Day
5	Month
6	Year - 1900

## Appendix C. Driver Error Messages

Message	Explanation
<b>SKgwcmd: #1</b> Error. Don't know this function=%d %x(h)	The value of the sk_function has been set incorrectly. The driver does not recognize it. <sup>§</sup> .
<b>SKgwcmd: #3.</b> Error. Too much data to send=%d max=%d.	The Silent Knight Gateway Command protocol can send a maximum of 7*255 bytes of data per message. Try breaking one Map Descriptor up so that less data is requested at one time. It may not be possible to set sk_module, sk_point, sk_zone or sk_group equal to zero as this requires too much data to be sent.
<b>SKgwcmd: #4.</b> Error. Cannot do Exp. Zone Stat. Mapdesc=<%s>	The command code is defined but the command and response format is not defined in the protocol specification. Change the sk_function parameter to another suitable command.
<b>SKgwcmd: #5</b> FYI. The mapDesc called <skgwcmd-stats> is too short.	Increase the value of the length parameter to at least 100. Read 0 for additional information.
<b>SKgwcmd: #6</b> Error. Dont know this function=%d %x(h)	See notes for error 1
<b>SKgwcmd: #7.</b> Error. Cannot do Exp. Zone Stat.	The command code is defined but the command & response format is not defined in the protocol specification.
<b>SKgwcmd: #8 Error</b> Incoming point status data from module=%d is being abandoned. MapDescriptors are required to define storage for. No further messages about data being abandoned will be provided.	If a Map Descriptor was defined which generated a poll for all point data then you require a Map Descriptor to specify where the point status data for each module must be stored. If this has not been done then this error message will be produced.
<b>SKgwcmd:#9</b> Error. Don't know this function=%d %x(h)	See notes for error 1.
<b>SKgwcmd:#11*</b> FYI. MapDesc Length too short rqd/actual=%d/%d MapDesc=<%s> Some incoming data will be abandoned	If a Map Descriptor generates a poll for all module points, zone or group data and the the SK device returns more data than provided for by the length parameter of the Map Descriptor then this error will be generated.
<b>SKgwcmd:#12*</b> FYI. Point Accessory Status Data abandoned. Define field=da_byte_name in mapDesc=<%s>	To eliminate this error, link the Map Descriptor to two Data Arrays, one for status data and the other for accessory data. See section 4.4.7..
<b>SKgwcmd:#13*</b> FYI. Data Array too short rqd/actual=%d/%d MapDesc=<%s> Some incoming data will be abandoned.	This error is similar to #11 but in this case the Data Array itself is too short.
<b>SKgwcmd:#14*</b> FYI. Point Accessory Status Data abandoned. Define field=da_byte_name in mapDesc=<%s>	See error #12.
<b>SKgwcmd:#15*</b> FYI. Port Checking Disabled ! If there is more than one SK device connected to this FieldServer this may produce errors.	If the FieldServer is to be connected to more than one Silent Knight device, the node definitions in the CSV file need to link to a port definition. Although normally not required for a Server side connection, this is not invalid.

<sup>§</sup> Edit the CSV file, download it to the FieldServer and reset the FieldServer for the changes to take effect.

\* Subsequent similar messages will be suppressed.

Message	Explanation
<b>SKgwcmd:#16*</b> FYI. Resetting/Silencing Array too short. <%s>	The value of the length parameter and the length of the Data Array conflict. §
<b>SKGWCMD:#17</b> Error. Incoming data is being abandoned. No further messages about data being abandoned will be provided.	Same error as #8 except that the driver produces this error in a different context.
<b>SKGWCMD:#18*</b> Error. Requested too much data from mapDesc=<%s> rqstd/avail: start=%d/%d. len=%d/%d.	More Zone Status data is contained in the response than the Map Descriptors length or starting zone number allows the driver to process. This error is produced by the Client driver. §
<b>SKgwcmd:#19 Error.</b> Incoming zone status data being abandoned. MapDesc's are required to define storage. No further messages about data being abandoned will be provided.	More Zone Status data is contained in the response than the Map Descriptors length or starting zone number allows the driver to process. This error is produced by the Server driver. §
<b>SKgwcmd:#20 FYI*</b> . Data Array too short rqd/actual=%d/%d MapDesc=<%s> Some incoming data will be abandoned.	This message is produced by the Server driver. It arises while processing zone status data. The message indicates that some data is being stored correctly but that 'surplus' data is being abandoned. Correct the length or starting zone number in the CSV file. §
<b>SKgwcmd:#21*</b> FYI. Data Array too short rqd/actual=%d/%d MapDesc=<%s> Some incoming data will be abandoned.	This message is produced by the Server driver. It arises while processing group status data. The message indicates that some data is being stored correctly but that 'surplus' data is being abandoned. Correct the length or starting zone number in the CSV file. §
<b>SKGWCMD:#22</b> Error. Incoming group status data being abandoned. MapDesc's are required to define storage. No further messages about data being abandoned will be provided.	This message is produced by the Server driver when the sk_group is set to zero and the Map Descriptor length is too small to store the incoming group data. Add a passive Map Descriptor to define the storage of the incoming data. §
<b>SKgwcmd:#23*</b> FYI. Additional data was discarded. You could have defined storage using the <da_byte_name> parameter in MapDesc=<%s>.	Group Status data consists of the group state and the number of troubles. The group state will be stored in the Data Array linked to the Map Descriptor. The number of troubles will be discarded unless the Map Descriptor is connected to a second Data Array using the da_byte_name parameter. This message is a warning only and no action is necessarily required.
<b>SKGWCMD:#24*</b> Error. Requested too much data from mapDesc=<%s> rqstd/avail: start=%d/%d. len=%d/%d.	This message is produced by the Client driver. It arises while processing group status data. The message indicates that some data is being stored correctly but that 'surplus' data is being abandoned. Correct the length or starting zone number. §
<b>SKgwcmd:#25</b> FYI. You could have used a MapDesc called <skgwcmd-stats> to expose diagnostic info.	This is a warning message only. No action is required. Read 0 for additional information.

§ Edit the CSV file, download it to the FieldServer and reset the FieldServer for the changes to take effect.

### Appendix C.1. Driver Stats

In addition to the standard FieldServer communication statistics described in the FieldServer Configuration Manual the DNP 3 Driver can also expose some driver statistics by writing data to a Data Array. A special Map Descriptor is required. The driver recognizes the Map Descriptor by its name which must be "**skgwcmd-stats**". The following example shows how this special Map Descriptor can be configured.

**Example**

Nodes			
Node_name, null_node,	Protocol SKgwcmd		
Data_Arrays			
Data_Array_Name, SK_STATS,	Data_Format, UINT32,	Data_Array_Length 300	
Map_blocks			
Map_block_Name, Skgwcmd-stats,	Data_Array_Name, SK_STATS,	Node_name, null_node,	length 300

When the driver sees this Map Descriptor it uses the Data Array SK\_STATS (in this example) to store driver specific statistics. Only one of these Map Descriptors may be specified per FieldServer.

The driver stores the following data.

Array Element	Contents
0	Not Used
1	SK_STAT_COMPL_HDR1
2	SK_STAT_COMPL_HDR2
3	SK_STAT_COMPL_CMD
4	SK_STAT_COMPL_BLK
5	SK_STAT_COMPL_CRC
6	SK_STAT_COMPL_CRC1
7	SK_STAT_COMPL_CRC2
8	SK_STAT_COMPL_BLK2
9	SK_STAT_MSTR_BAD
10	SK_STAT_MSTR_GOOD
11	SK_STAT_COMPL_SLV
12	SK_STAT_COMPL_GWAY
12	SK_STAT_SEND_MST_POLL
14	SK_STAT_SEND_SLV_IGNORE
15	SK_STAT_MST_IGNORE_GROUP
16	SK_STAT_MST_IGNORE_ZONE
17	SK_STAT_MST_IGNORE_POINT
18	SK_STAT_SEND_SLV_MSG
19	SK_STAT_SEND_SLV_MSG_BYTES
20	SK_STAT_SEND_SLV_CONCT_MSG
21	SK_STAT_SEND_SLV_RCVD_MSG
22	SK_STAT_SEND_SLV_RCVD_BYTES

THIS PAGE INTENTIONALLY LEFT BLANK