



A Sierra Monitor Company

Driver Manual
(Supplement to the FieldServer Instruction Manual)

FS-8704-25 Schneider Electric Cassia

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after November 2012

Driver Version: 1.00
Document Revision: 3

TABLE OF CONTENTS

1 SCHNEIDER ELECTRIC CASSIA Description..... 3

2 Driver Scope of Supply 3

 2.1 Supplied by FieldServer Technologies for this driver3

 2.2 Provided by the Supplier of 3rd Party Equipment3

 2.2.1 *Required 3rd Party Hardware*3

3 Hardware Connections..... 4

 3.1 Hardware Connection Tips / Hints.....4

4 Data Array Parameters..... 5

5 Configuring the FieldServer as a SCHNEIDER ELECTRIC CASSIA Client 5

 5.1 Client Side Connection Parameters5

 5.2 Client Side Node Parameters6

 5.3 Client Side Map Descriptor Parameters.....6

 5.3.1 *FieldServer Related Map Descriptor Parameters*6

 5.3.2 *Driver Related Map Descriptor Parameters*.....7

 5.3.3 *Timing Parameters*.....7

 5.4 Map Descriptor Example – How to read and write the Cassia Server Configuration Bits8

 5.5 Map Descriptor Example – How to setup Room Devices and Passively monitor them for data changes9

 5.6 Map Descriptor Example – How to actively monitor Room Devices for data changes in addition to passive monitoring10

 5.7 Map Descriptor Example – How to dump the Cassia Server’s Hierarchical Configuration to a file for configuration purposes.....11

Appendix A. Useful Features..... 13

Appendix B. Troubleshooting..... 13

 Appendix B.1. Driver Messages13

Appendix C. Vendor information 15

Appendix D. Reference 15

 Appendix D.1. The configuration status bits15

 Appendix D.2. List of Device Passive map descriptor types16

 Appendix D.3. Data Array Storage for Room_Status17

1 SCHNEIDER ELECTRIC CASSIA DESCRIPTION

The Schneider Electric Cassia driver allows the FieldServer to transfer data to and from a Cassia Building Management System Server over Ethernet using the Schneider Electric Cassia 3rd party socket interface protocol. The FieldServer can emulate only a Client.

Max Connections Supported

FieldServer Mode	Connections	Comments
Client	1	Connection to only 1 Cassia Server supported

2 DRIVER SCOPE OF SUPPLY

2.1 Supplied by FieldServer Technologies for this driver

FieldServer Technologies PART #	Description
FS-8915-10	UTP cable (7 foot) for Ethernet connection
FS-8915-10	UTP cable (7 foot) for RS-232 use
FS-8917-02	RJ45 to DB9F connector adapter
FS-8917-01	RJ45 to DB25M connection adapter

2.2 Provided by the Supplier of 3rd Party Equipment

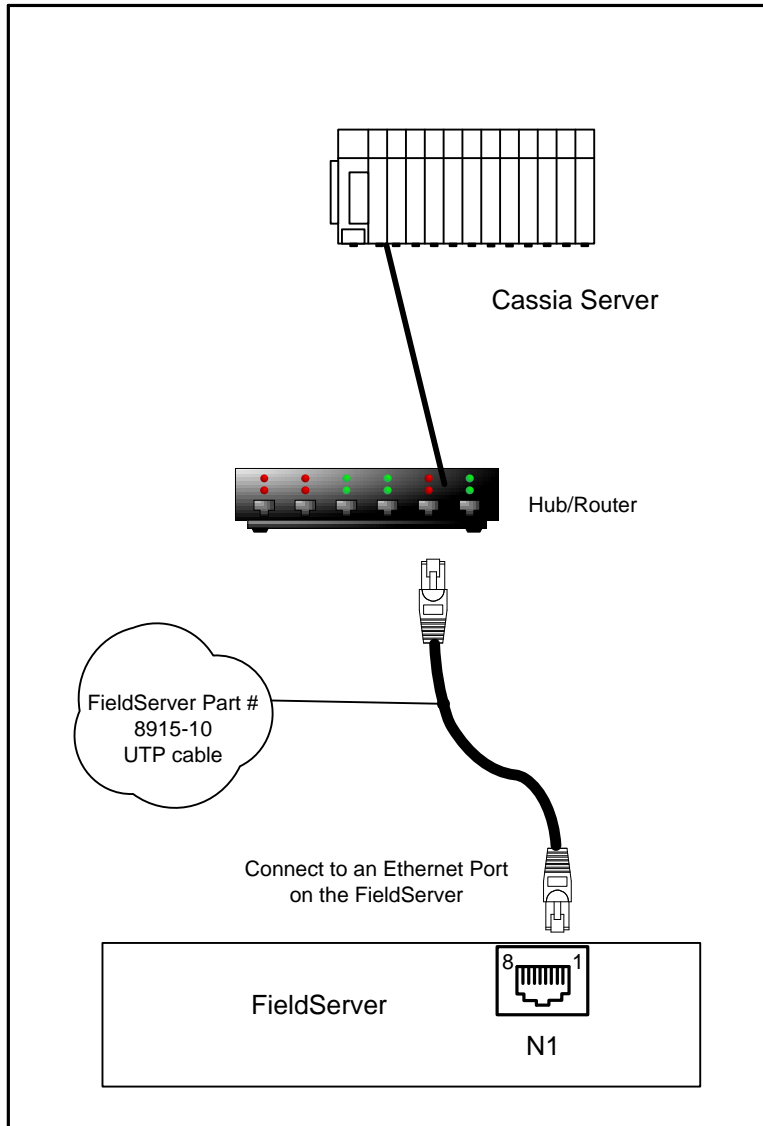
2.2.1 Required 3rd Party Hardware

Part #	Description
	Cassia Server

3 HARDWARE CONNECTIONS

The FieldServer is connected to the remote Cassia Server as shown in the connection drawing.

Configure the Cassia Server according to manufacturer's instructions



3.1 Hardware Connection Tips / Hints

None.

4 DATA ARRAY PARAMETERS

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

Section Title		
Data_Arrays		
Column Title	Function	Legal Values
Data_Array_Name	Provide name for Data Array	Up to 15 alphanumeric characters
Data_Array_Format	Provide data format. Each Data Array can only take on one format.	FLOAT, BIT, UInt16, Sint16, 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_Array_Format , Data_Array_Length
Config          , Bit                , 34
R1_States      , UInt16              , 2
```

5 CONFIGURING THE FIELDSEVER AS A SCHNEIDER ELECTRIC CASSIA 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 Cassia Server.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for Schneider Electric Cassia 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.

5.1 Client Side Connection Parameters

Section Title		
Connections		
Column Title	Function	Legal Values
Adapter	Specify which Ethernet port on the FieldServer to use	N1, N2
Protocol	Specify protocol used	Schneider Cassia, Schneider Electric Cassia EMS
EMS_Login_Enabled	Specify whether to login to the remote Cassia Server (not supported at this time)	Yes, No
EMS_Server_IP_Address	The remote IP address of the Cassia Server	Class A, B, C IP address, eg. 10.0.0.1

Example

```
// Client Side Connections

Connections
Adapter , Protocol , EMS_Login_Enabled , EMS_Server_IP_Address
N1 ,Schneider Cassia , No , 10.0.0.1
```

5.2 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	Cassia Server Config Node or Room ID for each room to be monitored on the remote Cassia Server	0 (for Config Node) 1-65534 (Room ID)
Protocol	Specify Protocol used	Schneider Cassia, Schneider Electric Cassia EMS
Adapter	Specify the Ethernet Adapter of the defined connection	N1, N2

Example

```
// Client Side Nodes

Nodes
Node_Name , Node_ID , Protocol , Adapter
Config , 0 , Schneider Cassia , N1
Room1 , 1 , Schneider Cassia , N1
Room7 , 7 , Schneider Cassia , N1
```

5.3 Client Side Map Descriptor Parameters

5.3.1 FieldServer Related Map Descriptor Parameters

Column Title	Function	Legal Values
Map_Descriptor_Name	Name of this Map Descriptor	Up to 32 alphanumeric characters
Data_Array_Name	Name of Data Array where data is to be stored in the FieldServer	One of the Data Array names from Section 4.
Data_Array_Offset	Starting location in Data Array	0 to (Data_Array_Length-1) as specified in Section 4.
Function	Function of Client Map Descriptor	RDB, RDBC, Passive depending on map descriptor application (see map descriptor examples)
Block_Type	Sets the write-through data-type on the Configuration Bits map descriptor	Flag (1-bit), Register (16-bits)

5.3.2 Driver Related Map Descriptor Parameters

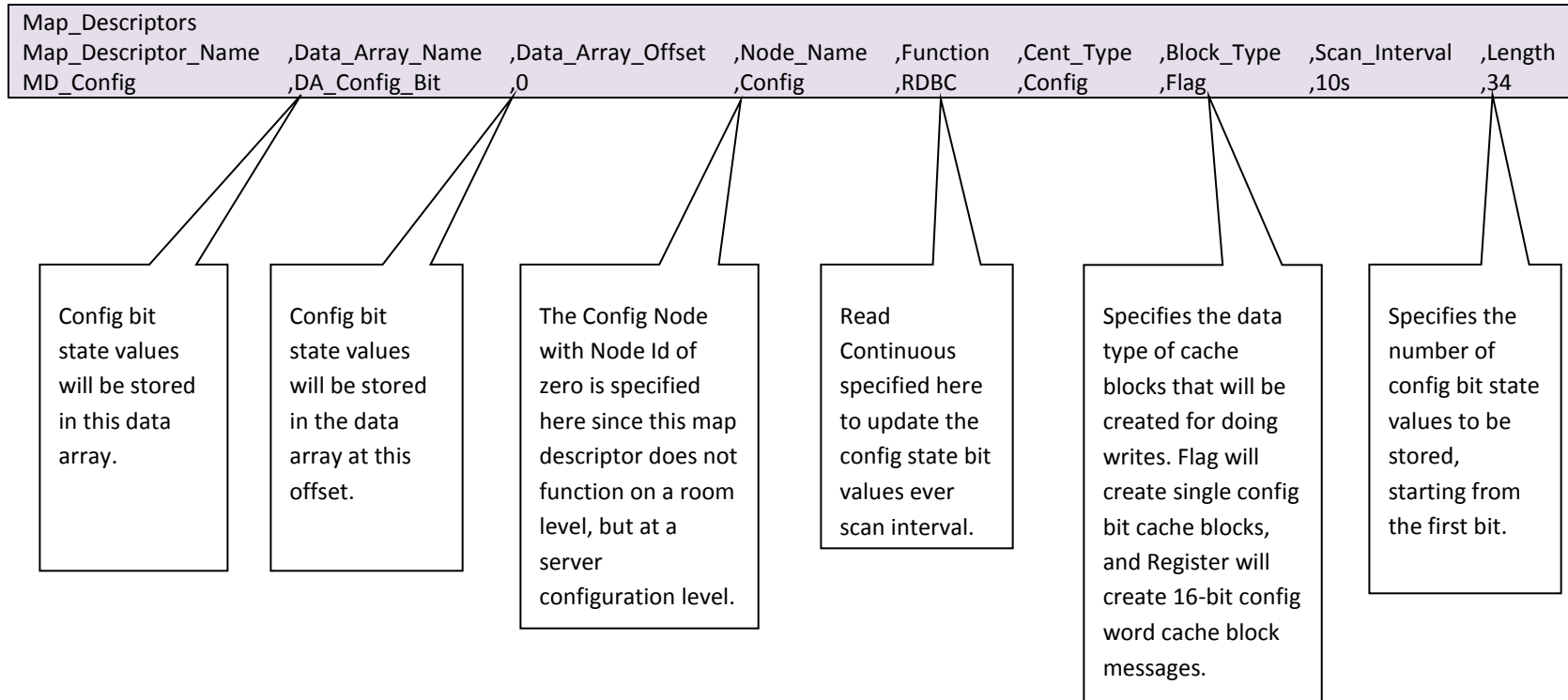
Column Title	Function	Legal Values
Node_Name	Name of Node to communicate with	One of the Node names specified in Section 5.2
Cent_Type	Specifies the type of message this map descriptor will send to or capture from the Cassia Server. Maximum length values are bracketed.	(Normal Use): Config (34-bit, 3-word) Room_Devices (1) Room_Status (2) Coordinator (3) Room_Coordinator (1) Thermostat (22) Door (3) Sliding_Door (3) Motion_Detector (3) Dimmer_Load (4) 1Button_Relay (4) 3Button_Relay (4) (Configuration Dump): Location_Dump (1) Floor_Dump (1) Room_Dump (1) Device_Dump (1)
Cent_Location_ID	Location Identifier	1-65534
Cent_Floor_ID	Floor Identifier (only used for Cassia Server configuration dump)	1-65534
Cent_Room_ID	Room Identified (only used for Cassia Server configuration dump)	1-65534
Length	Length of Map Descriptor which indicates the number of data elements to be stored	Greater or equal than 1 Refer to the maximum length values indicated in brackets in the Cent_Type legal values column.

5.3.3 Timing Parameters

Column Title	Function	Legal Values
Scan_Interval	Rate at which data is read from the Cassia Server	≥0.001s

5.4 Map Descriptor Example – How to read and write the Cassia Server Configuration Bits

For bit data array use:



For word data array use:

Map_Descriptors	Map_Descriptor_Name	,Data_Array_Name	,Data_Array_Offset	,Node_Name	,Function	,Cent_Type	,Block_Type	,Scan_Interval	,Length
MD_Config	,DA_Config_Uint16	,0	,Config	,RDBC	,Config	,Register	,10s	,3	

To change configuration bits on the Cassia Server, simply change the corresponding bit value or the complete word values in the data array defined for this map descriptor and a temporary write map descriptor called a cache block will be created by the driver to do the write to the remote Cassia Server.

Please see Appendix D.1 for the configuration status bits table.

5.5 Map Descriptor Example – How to setup Room Devices and Passively monitor them for data changes

Device Ids of remote devices configured in the Cassia Server are unique values that have to be retrieved and assigned by the SCHNEIDER ELECTRIC CASSIA driver to predefined device map descriptors.

For each room to be monitored, there has to be Room_Devices map descriptor defined with a RDB function so it only reads the room’s configured device Ids once and then assign them to predefined Passive map descriptors of the device types reported by the Cassia Server. Here is an example of a Room_Devices Map descriptor to retrieve the Device Ids for room 7. Note that a node with Node ID equal to 7 for Room7 had to be defined beforehand under the Nodes section.

Map_Descriptor_Name	,Data_Array_Name	,Data_Array_Offset	,Node_Name	,Function	,Cent_Type	,Length
MD_Room_7	,R7_Device_IDs	,0	,Room7	,RDB	,Room_Devices	,1

The Device Ids of all the devices configured in Room 7 will be stored to this data array.

The only function allowed here is RDB since the retrieval of Device Ids are only done once at startup.

The length of 1 does not affect the number of Device Ids stored. The map descriptor will store as many as the data array length allows and print a warning message if it ran out of space, so make sure the data array on this map descriptor is long enough for the expected number of devices in the room.

The second step is to add Passive function map descriptors for each device that will be reported by the Cassia Server. See the section called Useful Features in this manual on how to obtain a dump of the devices configured in a room from the Cassia Server if this is not known beforehand.

Here is an example of a Door device passive map descriptor:

Map_Descriptors							
Map_Descriptor_Name	,Data_Array_Name	,Data_Array_Offset	,Node_Name	,Function	,Cent_Location_ID	,Cent_Type	,Length
PMD_Door1_R7	,DA_Door1_R7	,0	,Room7	,Passive	,1	,Door	,3

At this stage the configuration is complete and the SCHNEIDER ELECTRIC CASSIA driver will receive unsolicited event messages from the Cassia Server for all device type that have been enabled for event reporting using the Config status bits map descriptor.

Device values per room can also be actively polled for by adding additional active map descriptors to effect the polling. See the next map descriptor example on how to actively monitor room devices in addition to receiving event messages. Passive event message receiving are always enabled on the driver, regardless of whether active room device map descriptors are defined or not. Device data will always be stored on the Passive function map descriptors, both for event data and active data.

Please see Appendix D.2 for the list of Device Passive map descriptor types and what values they store in their Data Array

5.6 Map Descriptor Example – How to actively monitor Room Devices for data changes in addition to passive monitoring

Map_Descriptors								
Map_Descriptor_Name	,Scan_Interval	,Data_Array_Name	,Data_Array_Offset	,Node_Name	,Function	,Cent_Location_ID	,Cent_Type	,Length
MD_Room1_Status	,10s	,Room1_States	,0	,Room1	,RDBC	,1	,Room_Status	,2

The room rented and occupancy state values are stored in this data array, and all other data are stored in the data arrays of the passive map descriptors of each device type reported by the Cassia Server.

The Node ID of the Node specified here is the Room ID number that will be polled for data.

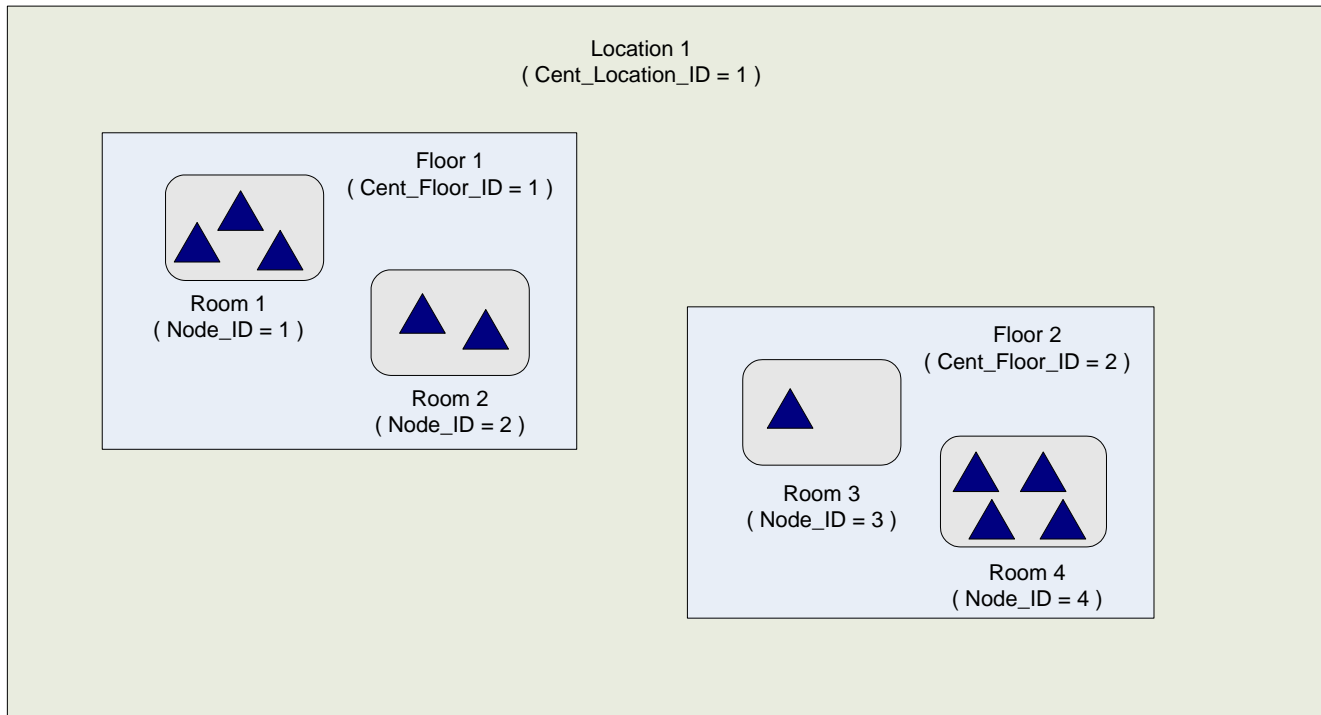
The room rented and occupancy state values are stored for a length of 2. A length of 1 will only store the rented state of the room.

Please see Appendix D.3 for Data Array Storage for Room_Status table

5.7 Map Descriptor Example – How to dump the Cassia Server’s Hierarchical Configuration to a file for configuration purposes

The Cassia Server’s hierarchical configuration can be dumped to a text file if it is not known beforehand. The dump information can then be used to create a FieldServer configuration file that has map descriptors for all rooms and devices to prevent event data not being stored. The SCHNEIDER ELECTRIC CASSIA driver will report any missing or extra map descriptors during the startup configuration step of the driver where it queries the Cassia Server for Device Ids per room to assign them to pre-defined Passive map descriptors of the corresponding device types.

The following diagram explains the hierarchical representation of locations, floors, rooms and devices configured in the Cassia Server:



To dump the information follow these steps:

1) Define the following map descriptor to retrieve the number of locations and their Ids:

Map_Descriptors							
Map_Descriptor_Name	,Data_Array_Name	,Data_Array_Offset	,Node_Name	,Function	,Cent_Type	,Length	
MD_Location_Dump	,-	,-	,Config	,RDB	,Location_Dump	,1	

Note that the Config Node should have a node Id of zero. A message on the RuiNet message screen will indicate when the location dump is complete. Upload and view the emsdump.txt file which will show the Location Ids configured on the Cassia Server.

2) Using the Location Ids from the previous step, define map descriptors with matching Cent_Location_ID values to retrieve the Floor configuration information from the Cassia Server:

Map_Descriptors							
Map_Descriptor_Name	,Data_Array_Name	,Data_Array_Offset	,Node_Name	,Function	,Cent_Location_ID	,Cent_Type	,Length
MD_Floor_Dump	,-	,-	,Config	,RDB	,1	,Floor_Dump	,1

3) Using the Floor Ids from the previous step, define map descriptors with matching Cent_Floor_ID values to retrieve the Room configuration information from the Cassia Server:

Map_Descriptors							
Map_Descriptor_Name	,Data_Array_Name	,Data_Array_Offset	,Node_Name	,Function	,Cent_Floor_ID	,Cent_Type	,Length
MD_RoomF1_Dump	,-	,-	,Config	,RDB	,1	,Room_Dump	,1
MD_RoomF2_Dump	,-	,-	,Config	,RDB	,2	,Room_Dump	,1

4) Using the Room Ids obtained from the previous step, define map descriptors with matching Cent_Room_ID values to retrieve the Device configuration information from the Cassia Server. Following this step, all information in the emsdump.txt file can be used to create the necessary Passive device map descriptors for storing event data.

Map_Descriptors							
Map_Descriptor_Name	,Data_Array_Name	,Data_Array_Offset	,Node_Name	,Function	,Cent_Room_ID	,Cent_Type	,Length
MD_DeviceF1R1_Dump	,-	,-	,Config	,RDB	,1	,Device_Dump	,1
MD_DeviceF1R2_Dump	,-	,-	,Config	,RDB	,2	,Device_Dump	,1
MD_DeviceF2R3_Dump	,-	,-	,Config	,RDB	,3	,Device_Dump	,1
MD_DeviceF2R4_Dump	,-	,-	,Config	,RDB	,4	,Device_Dump	,1

Appendix A. USEFUL FEATURES

The SCHNEIDER ELECTRIC CASSIA driver supports an information dump feature that can be used to query the hierarchical configuration of locations, floors, rooms and devices configured on a remote Cassia Server. The configuration information is dumped to a file called emsdump.txt that can be uploaded with the RuiNet utility and viewed (*Refer to Enote0096 for installing and operating the RuiNet Utility*). The information in the dumpfile can then be used to create a custom configuration file that matches the configuration in the Cassia Server. See the map descriptor example (Section 5.7) for more information.

Appendix B. TROUBLESHOOTING

The most common problem that may be encountered with the SCHNEIDER ELECTRIC CASSIA driver is missing or mismatched device map descriptors to store data from the Cassia Server. The driver will print out messages about missing map descriptors which can be viewed with the RuiNet utility. The user can then add these map descriptors to the FieldServer configuration file to allow data storage to be successful.

Appendix B.1. Driver Messages

Error #	Msg Screen	Screen message	Meaning	Suggested Solution
1	ERROR	DRV->CASSIA : Error, set config value of %d out of range, [0-33] cannot do write!	Config value bit offset can only be from 0 to 33	Make sure bit offset when triggering write is in range
2	ERROR	DRV->CASSIA : Error, remote server IP not defined on connection!	The remote Cassia server's IP address has not been defined on the connection	Define the IP address using the EMS_Server_IP_Address connection parameter
3	ERROR	DRV->CASSIA : Could not store assigned device ID to data array %s at offset %u, too short!	Data array is too short to store information	Increase data array length in config file
4	DRIVER	CASSIA : Add a passive mapdesc of this device type to your configuration for auto device id assignment!	A preceding message will indicate that a message for a certain type was received, but there is no Map Descriptor associated with the device	Add a device Map Descriptor to the configuration
5	DRIVER	CASSIA : Cannot store event for room id %u, node not defined!	A room event message was received, but there is no node associated	Add a node associated with the room ID to the configuration
6	DRIVER	CASSIA : Could not find door device id %u mapdesc for room id %u	Missing Map Descriptor from configuration	Add Map Descriptor to configuration
7	DRIVER	CASSIA : Could not find motion detector device id %u mapdesc for room id %u	Missing Map Descriptor from configuration	Add Map Descriptor to configuration
8	DRIVER	CASSIA : Could not find sliding door device id %u mapdesc for room id %u	Missing Map Descriptor from configuration	Add Map Descriptor to configuration
9	DRIVER	CASSIA : Could not find switch relay or dimmer device id %u mapdesc for room id %u	Missing Map Descriptor from configuration	Add Map Descriptor to configuration
10	DRIVER	CASSIA : Could not find thermostat device id %u	Missing Map Descriptor from configuration	Add Map Descriptor to configuration

		mapdesc for room id %u		
11	DRIVER	CASSIA : Discarding dimmer or relay button event for room id %u until room configured	An event was received for a room that is not configured	Add room to configuration or ignore if this event was not expected
12	DRIVER	CASSIA : Discarding dimmer or relay load event for room id %u until room configured	An event was received for a room that is not configured	Add room to configuration or ignore if this event was not expected
13	DRIVER	CASSIA : Discarding door event for room id %u until room configured	An event was received for a room that is not configured	Add room to configuration or ignore if this event was not expected
14	DRIVER	CASSIA : Discarding motion detected event for room id %u until room configured	An event was received for a room that is not configured	Add room to configuration or ignore if this event was not expected
15	DRIVER	CASSIA : Discarding sliding door event for room id %u until room configured	An event was received for a room that is not configured	Add room to configuration or ignore if this event was not expected
16	DRIVER	CASSIA : Discarding thermostat event for room id %u until room configured	An event was received for a room that is not configured	Add room to configuration or ignore if this event was not expected
17	DRIVER	CASSIA : Discarding thermostat temp event for room id %u until room configured	An event was received for a room that is not configured	Add room to configuration or ignore if this event was not expected
18	DRIVER	CASSIA : Error, cannot find coordinator mapdesc to store async event data ..	Missing Map Descriptor from configuration	Add Map Descriptor to configuration
19	DRIVER	CASSIA : Error, cannot find dimmer relay button mapdesc to store async event data ..	Missing Map Descriptor from configuration	Add Map Descriptor to configuration
20	DRIVER	CASSIA : Error, cannot find dimmer relay load mapdesc to store async event data ..	Missing Map Descriptor from configuration	Add Map Descriptor to configuration
21	DRIVER	CASSIA : Error, cannot find door mapdesc to store async event data ..	Missing Map Descriptor from configuration	Add Map Descriptor to configuration
22	DRIVER	CASSIA : Error, cannot find motion sensor mapdesc to store async event data ..	Missing Map Descriptor from configuration	Add Map Descriptor to configuration
23	DRIVER	CASSIA : Error, cannot find room id %d status mapdesc	Missing Map Descriptor from configuration	Add Map Descriptor to configuration
24	DRIVER	CASSIA : Error, cannot find sliding door mapdesc to store async event data ..	Missing Map Descriptor from configuration	Add Map Descriptor to configuration
25	DRIVER	CASSIA : Error, cannot find thermostat mapdesc to store async event data ..	Missing Map Descriptor from configuration	Add Map Descriptor to configuration
26	DRIVER	CASSIA : Error, cannot find thermostat mapdesc to store async event data2 ..	Missing Map Descriptor from configuration	Add Map Descriptor to configuration
27	DRIVER	CASSIA : Error, could not assign device id %u from room id %u to a mapdesc!	Missing Map Descriptor from configuration	Add Map Descriptor to configuration

28	DRIVER	CASSIA : Error, could not find room id %d devices mapdesc!	Missing Map Descriptor from configuration	Add Map Descriptor to configuration
29	DRIVER	CASSIA : No server response to ping	The Cassia server cannot be reached over the Ethernet	Check that server is running and firewall port 10042 opened
30	DRIVER	CASSIA : Please add a passive mapdesc of device type %u	Missing Map Descriptor from configuration	Add Map Descriptor to configuration
31	DRIVER	CASSIA : Received an unknown command 0x%x	A message was received that is not known to the driver	Take a FST_Diagnostic capture and send it to Technical Support
32	DRIVER	CASSIA : Error, could not find config mapdesc to store data!	Missing Map Descriptor from configuration	Add Map Descriptor to configuration

Appendix C. VENDOR INFORMATION

The SCHNEIDER ELECTRIC CASSIA driver connects to remote TCP port 10042 of the Cassia Server. The driver keeps the TCP/IP connection active by sending a Cassia Server ping command every 45 seconds should there not have been an active poll from a map descriptor during this time.

Appendix D. REFERENCE

Appendix D.1. The configuration status bits

The configuration status bits are defined as follows with respect to their enabling or disabling events reporting from the Cassia Server:

Offset into Data Array	Config value to set
0	Enable Send Thermostat Temperature Changed Events
1	Disable Send Thermostat Temperature Changed Events
2	Enable Send Thermostat Temperature Changed Variables Events
3	Disable Send Thermostat Temperature Changed Variables Events
4	Enable Send Thermostat Setpoint Changed Events
5	Disable Send Thermostat Setpoint Changed Events
6	Enable Send Door Opened Events
7	Disable Send Door Opened Events
8	Enable Send Door Closed Events
9	Disable Send Door Closed Events
10	Enable Send Motion Detected Events
11	Disable Send Motion Detected Events
12	Enable Send Sliding Door Opened Events
13	Disable Send Sliding Door Opened Events
14	Enable Send Sliding Door Closed Events
15	Disable Send Sliding Door Closed Events
16	Enable Send Dimmer / Relay Load On Events
17	Disable Send Dimmer / Relay Load On Events
18	Enable Send Dimmer / Relay Load Off Events
19	Disable Send Dimmer / Relay Load Off Events
20	Enable Send Dimmer / Relay Button Released Events

21	Disable Send Dimmer / Relay Button Released Events
22	Enable Send Dimmer / Relay Button Pressed Events
23	Disable Send Dimmer / Relay Button Pressed Events
24	Enable Send Dimmer / Relay Button Held Events
25	Disable Send Dimmer / Relay Button Held Events
26	Enable Send Dimmer / Relay Button Double Tap Events
27	Disable Send Dimmer / Relay Button Double Tap Events
28	Enable Send Coordinator Connected Events
29	Disable Send Coordinator Connected Events
30	Enable Send Coordinator Disconnected Events
31	Disable Send Coordinator Disconnected Events
32	Enable All Events
33	Disable All Events

Appendix D.2. List of Device Passive map descriptor types

List of Device Passive map descriptor types and what values they store in their Data Array

Cent_Type = Coordinator

Offset	Description
0	Coordinator ID
1	Coordinator type: 1 = ethernet, 2 = serial
2	Coordinator event: 0 = disconnected, 1 = connected

Cent_Type = Room_Coordinator

Offset	Description
0	None – not defined by protocol

Cent_Type = Thermostat

Offset	Description
0	major version
1	minor version
2	current setpoint
3	current setpoint cool
4	current setpoint heat
5	current humidity setpoint
6	current temperature in fahrenheit
7	current humidity
8	upper limit for setpoints
9	lower limit for setpoints
10	temperature display of thermostat, 1=celcius, 2=fahrenheit
11	set mode of thermostat, 10=off, 7=cool, 8=heat, 9=auto
12	current running mode of thermostat, 0=idle, 1=cooling, 2=heating, 3=2nd stage cooling, 4=2nd stage heating, 5=other
13	current fan status of thermostat, 0=idle, 1=on, 3=low, 4=medium, 5=high
14	set fan mode of thermostat, 12=on, 2=auto, 3=low, 4=medium, 5=high
15	differential
16	setback differential
17	deep setback differential
18	setback mode, 0=no setback, 1=setback, 2=deep setback
19	vip mode, 1=yes, 0=no

20	thermostat button pressed, 1=bottom.left, 2=bottom.middle, 3=bottom.right, 4=down, 5=up
21	thermostat button action, 0=none, 1=pressed

Cent_Type = Door, Sliding_Door, Motion_Detector

Offset	Description
0	battery level of sensor in millivolts, eg. 3000mV = 3.0 Volts
1	button status, 0=down, 1=up
2	doors: door state, 0=closed, 1=open ; motion detector: none=0, detected=1

Cent_Type = 1Button_Relay , 3Button_Relay

Offset	Description
0	load level in percentage, 0-100
1	load status, 1=on, 0=off
2	button event, 0=released, 1=pressed, 2=held, 3=doubletapped
3	button that caused event, 0=lower, 1=middle, 2=top ?

Appendix D.3. Data Array Storage for Room_Status

Offset	Description
0	Rented State of Room 0 = Unrented ; 1 = Rented ; 2 = Out of order ; 3 = Off Market ; 4 = Unknown
1	Occupancy State of Room 0 = Vacant ; 1 = Occupied ; 2 = Maid in Room