



## **FieldServer**

### **FS-8707-11 Convia XML**

#### **Driver Manual**

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

#### **APPLICABILITY & EFFECTIVITY**

Effective for all systems manufactured after June 2019.

Driver Revision: 1.01  
Document Revision: 1.C

## Technical Support

Please call us for any technical support needs related to the FieldServer product.

Sierra Monitor Corporation  
1991 Tarob Court  
Milpitas, CA 95035

Website: [www.sierramonitor.com](http://www.sierramonitor.com)

U.S. Support Information:

+1 408 964-4443

+1 800 727-4377

Email: [support@sierramonitor.com](mailto:support@sierramonitor.com)

EMEA Support Information:

+31 33 808 0590

Email: [support.emea@sierramonitor.com](mailto:support.emea@sierramonitor.com)

**TABLE OF CONTENTS**

- 1 Convia XML Description ..... 4**
- 2 Driver Scope of Supply ..... 4**
  - 2.1 Supplied by Sierra Monitor ..... 4
  - 2.2 Provided by the Supplier of 3<sup>rd</sup> Party Equipment..... 4
    - 2.2.1 Required 3<sup>rd</sup> Party Hardware..... 4
- 3 Hardware Connections..... 5**
- 4 Data Array Parameters ..... 6**
- 5 Configuring the FieldServer as a Convia XML Client..... 7**
  - 5.1 Client Side Connection Descriptors ..... 7
  - 5.2 Client Side Node Descriptors ..... 8
  - 5.3 Client Side Map Descriptors ..... 8
    - 5.3.1 FieldServer Related Map Descriptor Parameters ..... 8
    - 5.3.2 Driver Related Map Descriptor Parameters ..... 9
    - 5.3.3 Timing Parameters ..... 9
  - 5.4 Map Descriptor Example ..... 10
- 6 Configuring the FieldServer as a Convia XML Server ..... 11**
  - 6.1 Server Side Connection Descriptors ..... 11
  - 6.2 Server Side Node Descriptors ..... 11
  - 6.3 Server Side Map Descriptors..... 11
  - 6.4 FieldServer XML Schema..... 12
    - 6.4.1 Root Element..... 12
    - 6.4.2 Child Elements ..... 13
    - 6.4.3 Sub-Child Elements..... 13
  - 6.5 Using HTTP-GET to Read FieldServer Data..... 14
  - 6.6 Using HTTP-POST to Write Data to the FieldServer ..... 15
- Appendix A. Reference..... 16**
  - Appendix A.1. Sending Commands to the Gateway ..... 16
    - Appendix A.1.1. Sending Device Commands..... 16
    - Appendix A.1.2. Control Commands ..... 16
  - Appendix A.2. Reading Device Data File ..... 17
    - Appendix A.2.1. Using GetFieldServerXML ..... 17
    - Appendix A.2.2. Using GetZoneDeviceListXML ..... 17
  - Appendix A.3. Interpreting Read Data..... 18
    - Appendix A.3.1. Typical Results from Read ..... 18
    - Appendix A.3.2. Attributes ..... 18
    - Appendix A.3.3. Device Class ..... 18
    - Appendix A.3.4. HVAC Modes..... 19

## 1 CONVIA XML DESCRIPTION

The Convia driver uses XML syntax to communicate with the Convia Global Gateway device. As a Client the Convia driver reads specified data points from the Convia gateway and stores the data in the FieldServer Data Arrays. The Client driver achieves this by using XML element and attribute syntax. As a Server the Convia driver allows the Convia Global Gateway to use the FieldServer XML Schema syntax to read data from the FieldServer Data Arrays.

### Max Nodes Supported

FieldServer Mode	Nodes	Comments
Client	Practical limit of 20	The Convia driver places no restriction on how many Global Gateways can be accessed. Practically this number should be restricted to 20.
Server	1	The FieldServer as a Server device will appear as a single Server device to the Convia Global gateway.

## 2 DRIVER SCOPE OF SUPPLY

### 2.1 Supplied by Sierra Monitor

Part #	Description
FS-8915-10	UTP cable (7 foot) for Ethernet connection <sup>1</sup>

### 2.2 Provided by the Supplier of 3<sup>rd</sup> Party Equipment

#### 2.2.1 Required 3<sup>rd</sup> Party Hardware

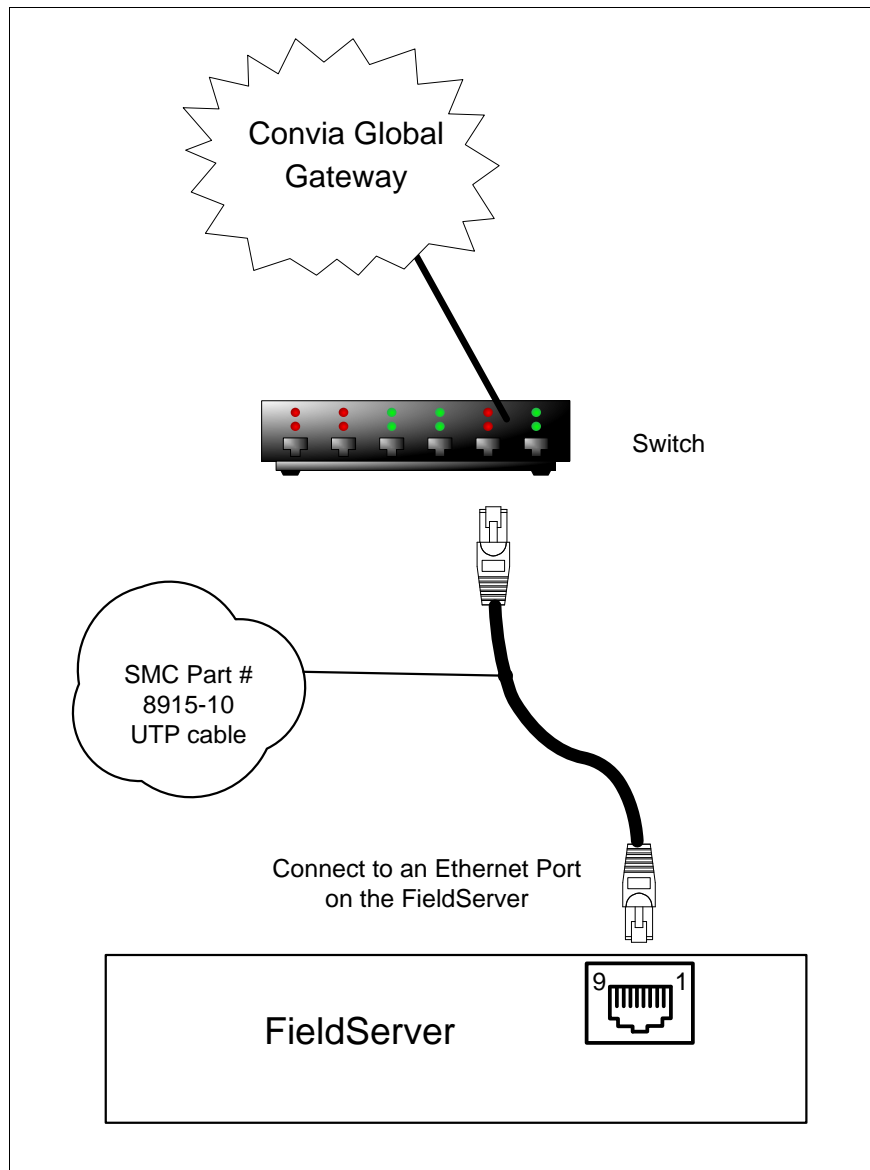
Part #	Description
	Ethernet 10/100 BaseT witch <sup>2</sup>

<sup>1</sup> This cable is necessary for connection to the driver. It is shipped with the FieldServer and not separately with the driver.

<sup>2</sup> Not all FieldServer models support 100BaseT. Consult the appropriate instruction manual for details of the Ethernet speed supported by specific hardware.

### 3 HARDWARE CONNECTIONS

The FieldServer is connected to the Convia Global Gateway as shown in connection drawing. Configure the Convia Global Gateway according to manufacturer's instructions.



## 4 DATA ARRAY PARAMETERS

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

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

**Example**

```

// Data Arrays
Data_Arrays
Data_Array_Name , Data_Array_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 CONFIGURING THE FIELDSEVER AS A CONVIVA XML CLIENT

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

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a Conviva XML Server.

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

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

### 5.1 Client Side Connection Descriptors

Section Title		
Connections		
Column Title	Function	Legal Values
Connection	Connection Name	N1, N2 <sup>3</sup>
Protocol	Specify protocol used.	XML_Convia
Poll_Delay*	Time between internal polls.	0-32000s; <b>0.05s</b>

**Example**

```
// Client Side Connections

Connections
Connection    , Protocol
N1            , XML_Convia
```

<sup>3</sup> Consult the appropriate Instruction manual for details of the ports available on specific hardware.

## 5.2 Client Side Node Descriptors

Section Title		
Nodes		
Column Title	Function	Legal Values
Node_Name	Provide name for the Node.	Up to 32 alphanumeric characters
Node_ID*	Identification number for this Node. Not required to communicate to the Convia Gateway, but required if Node_Status Data_Arrays are defined. Refer to the FieldServer configuration Manual for more information.	Any 32-bit value, -
Protocol	Specify Protocol used.	XML_Convia
Connection	Specify through which port the device is connected to the FieldServer.	N1, N2 <sup>3</sup>

### Example

```
// Client Side Nodes

Nodes
Node_Name , Node_ID, , Protocol , Connection
Gateway_1 , 1 , XML_Convia , N1
```

## 5.3 Client Side Map Descriptors

### 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 Data Array section
Data_Array_Offset	Starting location in Data Array.	0 to (Data_Array_Length-1) as specified in Data Array section
Function	Function of Client Map Descriptor.	RDBC, Passive_Client



### 5.3.2 Driver Related Map Descriptor Parameters

Column Title	Function	Legal Values
Node_Name	Name of Node from which to fetch data.	One of the Node names specified in Client Node Descriptor
Command	Specify the command method used to fetch the XML data.	HTTP-GET
XML-URL	Specifies an actual "file" to fetch from the remote server device.	Any valid string; Refer to <a href="#">Appendix A.2</a> for more information
Attribute	Specify the XML Attribute to read.	Refer to <a href="#">Appendix A.3.2</a> for a list of valid attributes
Search_Attribute	Specify the Attribute to be used to identify the XML entry.	Refer to <a href="#">Appendix A.3.2</a> for a list of valid attributes
Search_Value	Used in conjunction with the Search_Attribute to identify an XML entry.	Any valid string defined by the Device for the Search_Attribute used
Write_Command	When a write-thru occurs on the Map_Descriptor, the specified Write_Command string is used to perform the write.	A command string enclosed in ""; Refer to <a href="#">Appendix A.1</a> for more information

### 5.3.3 Timing Parameters

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

## 5.4 Map Descriptor Example

**At least one** Client Side Map Descriptor needs to be defined to select the actual XML page containing Attributes and Elements relating to the XML data on the Remote Gateway device. Multiple Map Descriptors can be configured to fetch different XML pages.

**NOTE: The function of the Map Descriptor must be Rdbc (see example below).**

```
// Client Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Command
CMD_DEV_DATA , DA_HTTP_STR , 0 , Rdbc , Gateway_1 , HTTP-GET

, XML-URL , Scan_Interval
, "cgi-bin/tscontrol?Command=GetFieldServerXML&" , 2s
```

Selected data from the XML page(s) selected above is extracted and placed into Data Arrays.

```
// Read from Convia Smart Controllers (Two controllers namely C1 and C2 are specified below)
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name
C1_ID_Number , DA_ID , 0 , Passive_Client , Gateway_1
C1_Volume , DA_VOLUME , 0 , Passive_Client , Gateway_1
C2_ID_Number , DA_ID , 1 , Passive_Client , Gateway_1
C2_Volume , DA_VOLUME , 1 , Passive_Client , Gateway_1
```

```
, Attribute , Search_Attribute , Search_Value , Write_Command
, Idnumber , Name , Lounge , -
, Volume , Name , Lounge , "Value=<local>&dataitem=<DA_ID:0>"
, Idnumber , Name , Dining , -
, Volume , Name , Dining , "Value=<local>&dataitem=<DA_ID:1>"
```

The Search Attribute and Search Value define the Attribute and Attribute Value to be used as the search criteria for finding the correct Child Element.

The <local> and <Data\_Array\_Name> values are replaced by actual values from the Data Arrays before the write is executed.

## 6 CONFIGURING THE FIELDSEVER AS A CONVIVA XML SERVER

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

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a Conviva XML Client.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for Conviva XML communications, the driver independent FieldServer buffers need to be declared in the “Data Arrays” section. No Nodes or Map Descriptors need to be configured for the Conviva XML Server. Data requests are performed using the HTTP GET command. Multiple Data Arrays can be requested at the same time. Data writes are performed using the HTTP POST command.

### 6.1 Server Side Connection Descriptors

Section Title		
Connections		
Column Title	Function	Legal Values
Connection	Connection Name	N1, N2 <sup>4</sup>
Protocol	Specify protocol used.	XML_Convia

#### Example

```
// Server Side Connections

Connections
Connection          , Protocol
N1                  , XML_Convia
```

### 6.2 Server Side Node Descriptors

No server side node descriptors must be configured.

### 6.3 Server Side Map Descriptors

No server side map descriptors must be configured.

<sup>4</sup> Consult the appropriate Instruction manual for details of the ports available on specific hardware.

## 6.4 FieldServer XML Schema

The FieldServer XML schema is arranged as a tree structure consisting of a Root Element, multiple Child Elements which define the Data\_Arrays in the FieldServer, and one Sub-Child element per Child Element which define the data in the Data Arrays as well as the attributes of the data (e.g. validity). Attributes are used to specify the parameters of the Elements in the tree.

**NOTE: The Element Name is ALWAYS in lowercase.**

The diagram shows the general format of the XML tree.

```

<Root Attribute="value" .. >
  <Child Attribute="value" .. >
    <Subchild Attribute="value" .. > value...</Subchild>
  </Child>
  <Child Attribute="value" .. >
    <Subchild Attribute="value" .. > value...</Subchild>
  </Child>
  ..
  ..
</Root>

```

### 6.4.1 Root Element

Element Name	Function	Legal Values
Data_Arrays	The root element is always present.	Only one root element can exist
Attributes	Function	Legal Values
XML_VERSION	Reports the version of the FieldServer XML Schema.	Takes the form of V.vva where V=Major version, vv=feature version and a=bugfix version. Changes in V or vv indicate whether additional elements or attributes have been added to the FieldServer XML schema.
BRIDGE_TITLE	Title String.	The Title keyword defined in the config.csv file under the comment section
Values	Function	Legal Values
None	Not Applicable.	No values are contained in the Root Element

### 6.4.2 Child Elements

Element Name	Function	Legal Values
Data_Array	Defines the Data Arrays present in the FieldServer.	One Child Element will be reported for each Data Array defined under the Data_Arrays section in the configuration file
Attributes	Function	Legal Values
NAME	Name of Data Array.	String of maximum 15 characters
FORMAT	Data Format of the Data Array.	FLOAT for floating point data, or BIT for binary data
LENGTH	Number of data items in this Data Array.	Integer value
MAX_INDEX	Identifies the maximum number of Data_Arrays on the FieldServer.	This value is not specified but is returned on polling the FieldServer
INDEX	Identifies the Data Array number. The first Data Array will be and Index of 1. The last Data Array will have an index of Max_Index.	Any positive integer <= MAX_INDEX
Values	Function	Legal Values
None	Not Applicable.	No values are contained in Child Elements

### 6.4.3 Sub-Child Elements

Element Name	Function	Legal Values
data	Contains the data in the Data Array.	Only one Sub-Child element exists for every Data Array defined
Attributes	Function	Legal Values
OFFSET	Defines start offset into the Data_Array.	Integer value > 0 and smaller than the length of the Data Array
DATA_AGE	Indicates how old the data was when the data child element was reported.	String indicated the time with 2 decimal places and terminated with the characters
STATUS	Indicates the status of the data.	0 = Data is valid >0 = Data is unreliable or invalid
Values	Function	Legal Values
Data Array value	Contains a list of values pertaining to the specified Data Array.	The legal values depend on the FORMAT of the Data Array and take the form of a space separated string of values

## 6.5 Using HTTP-GET to Read FieldServer Data

The HTTP-GET command is used to request data XML data from the FieldServer. The filename specified must be data\_arrays.xml. Multiple sets of parameters can be specified using the ? separator character. The Name, offset and length parameters are optional.

`data_arrays.xml?NAME=data_array_name:length.offset`

Parameter	Description	Legal Values
Data_Array_Name*	Specify the name of the Data Array.	One of the Data Array names from Data Array section. If not specified then all Data Arrays will be displayed.
Offset*	Starting location in Data Array.	0 to (Data_Array_Length-1) as specified in Data Array section, <b>Default 0</b>
Length*	Specify the length.	If the length is not specified, the full length will be used

### HTTP-GET Command

`data_arrays.xml?NAME=DA_AI_01:3.4&NAME=DA_BI_01`

### XML Response

```
<data_arrays XML_VERSION="1.00a" BRIDGE_TITLE="Convia example config">
  <data_array NAME="DA_AI_01" FORMAT="FLOAT" LENGTH="10" INDEX="1"
    MAX_INDEX="4">
    <data OFFSET="4" DATA_AGE=45.34s STATUS="0">4.32 12.56 12.45</data>
  </data_array>
  <data_array NAME=DA_BI_01 FORMAT="BIT" INDEX="1" MAX_INDEX="3">
    <data OFFSET="0" DATA_AGE=23.34s>1 0 1 0 1 1</data>
  </data_array>
</data_arrays>
```

## 6.6 Using HTTP-POST to Write Data to the FieldServer

The HTTP-POST command is used to write XML data to the FieldServer Data Arrays. The format of the HTTP command is described below.

*NAME=data\_array\_name&OFFSET=offset&VALUE=value*

Parameter	Description	Legal Values
Data_Array_Name*	Specify the name of the Data Array.	One of the Data Array names from Data Array section. If not specified then all Data Arrays will be displayed.
Offset*	Starting location in Data Array.	0 to (Data_Array_Length-1) as specified in Data Array section, <b>Default 0</b>
Value	Specify the Value to write.	Any floating point or SInt value

### HTTP-POST Command

Typically a web page is used to post commands. An example web page is presented below.

```
<html>
  <body>
    <form name="input" action="http://192.168.1.5"
      method="post">
      Data_Array_Name:
      <input type="text" name="NAME">
      <br/>
      Data_Array_Offset:
      <input type="text" name="OFFSET">
      <br/>
      Data_Array_Value:
      <input type="text" name="VALUE">
      <br/>
      <input type="submit" value="Submit">
    </form>
  </body>
```

### XML Response

If write succeeded:

```
<HTML><BODY>Done</BODY></HTML>\n
```

If write failed:

```
<HTML><BODY>ERROR:Invalid Parameters for Command </BODY></HTML>\n
```

## Appendix A. Reference

### Appendix A.1. Sending Commands to the Gateway

#### Appendix A.1.1. Sending Device Commands

The format for sending commands to control devices in the gateway is as follows:

```
/cgi-bin/tscontrol?Command=Control&What=<COMMAND>&Value=<VALUE>&dataitem=<ITEMNUM>&
```

Where:

**<COMMAND>** Is the command to send to control a device. A list of available controls is provided in

**<VALUE>** Is the value to send with some commands.

**<ITEMNUM>** Is the device ID Number that informs the gateway of which device to control.

#### Appendix A.1.2. Control Commands

Command	Description	Value Settings
DIM_ADD	Add to the dim level <VALUE>	
DIM_SUBTRACT	Subtract from the dim level <VALUE>	
DIM	Dim to specific level <VALUE>	
HEAT_ADD	Add to heat setpoint <VALUE>	
HEAT_SUBTRACT	Subtract from heat setpoint <VALUE>	
COOL_ADD	Add to Cool Setpoint <VALUE>	
COOL_SUBTRACT	Subtract from Cool Setpoint<VALUE>	
TOGGLE	Toggle the load between ON and OFF.	
SCENE1	Enable Scene 1	
SCENE2	Enable Scene 2	
SCENE3	Enable Scene 3	
SCENE4	Enable Scene 4	
AC_SET_HEAT	Set AC Heat setpoint to <VALUE>	
AC_SET_COOL	Set AC Cool setpoint to <VALUE>	
AC_COOL	Sets the HVAC into COOL mode.	
AC_HEAT	Sets the HVAC into HEAT mode.	
AC_AUTO	Sets the HVAC into AUTO mode.	
AC_OFF	Sets the HVAC into OFF mode.	
AC_FAN	Turns the fan on and off by setting <VALUE>	0=OFF, 1=ON
AC_HOLD	Turns the hold mode on by setting <VALUE>	0=OFF, 1=ON
ON	Turns the device on.	
OFF	Turns the device off.	
DIM	Sets the device to dim <VALUE>	

#### Example

```
"cgi-bin/tscontrol?Command=Control&What=DIM&Value=<local>&dataitem=<DA_ID_01:0>&"
```

**NOTE:** The write string can link values from any number of Data\_Arrays using the <> syntax. The <local> entry refers to the Map\_Descriptors own Data\_Array.



## Appendix A.2. Reading Device Data File

This file contains a list of devices and their current status.

There are currently two commands that can be used to read this file:

- GetFieldServerXML
- GetZoneDeviceListXML

### Appendix A.2.1. Using GetFieldServerXML

This command returns all the data for the FieldServer in one read. No parameters need to be defined.

The format to be used to read the file is as follows:

```
/cgi-bin/tscontrol?Command=GetFieldServerXML&
```

### Appendix A.2.2. Using GetZoneDeviceListXML

This command is used to receive the relevant information for devices in a selected zone based on USERID.

The format to be used to read the file is as follows:

```
/cgi-bin/tscontrol?Command=GetZoneDeviceListXML&ZoneNum=<CURRENTZONEID>&UserID=<USERID>&
```

Where:

**<CURRENTZONEID>** The ID number of the Zone containing the required data. If 0 is inserted, the gateway will go to the root zone for the user based on the USERID.

**<USERID>** The user ID number which determines the user's access to specified zones.

## Appendix A.3. Interpreting Read Data

### Appendix A.3.1. Typical Results from Read

```
<devices>
  <device name="Remote" room="/Top Level/Remote" type="Remote" icon="default.gif"
  idnumber="219" onstatusname="On" offstatusname="Off" brightstatusname="Bright"
  dimstatusname="Dim" volume="0" coolsetpoint="80" heatsetpoint="0" currenttemp="0"
  hvacmode="HEAT" hvacfan="0" hvachold="0" indexnumber="1"/>
</devices>
```

### Appendix A.3.2. Attributes

Attribute	Description
name	The name that is used to describe the device.
room	The zone path for the device in the hierarchy.
type	The device class. A list of device classes is presented in <a href="#">Appendix A.3.3</a> .
icon	The Icon name used for this device. The icon file can be retrieved from the gateway using the following address. /MCE/Icons/<ICONNAME>
idnumber	The ID Number used to describe the device on the gateway. This number is used later on for sending commands to the gateway.
onstatusname	The status name to be used when the device is on.
offstatusname	The status name to be used when the device is off.
brightstatusname	The status name to be used when the device was brightened.
dimstatusname	The status name to be used when the device was dimmed.
volume	The current volume or dim level setting.
coolsetpoint	The current cool setpoint used for both HVAC and HVACSingle.
heatsetpoint	The current heat setpoint used for only HVAC.
currenttemp	The current room temperature used for both HVAC and HVACSingle.
hvacmode	The current HVAC mode for HVAC only. A list of modes is presented in <a href="#">Appendix A.3.4</a> .
hvacfan	0 = fan is off and 1 = fan is on - available for HVAC only.
hvachold	0 = hold is off and any other value the hold is on - available for HVAC only.
indexnumber	The index number to refer to when controlling audio, events and cameras.

### Appendix A.3.3. Device Class

Device Class	Description
Dimmable	This is a dimmable lighting device.
Switch	This device only supports on and off commands.
Status	This device only reports the current status.
HVAC	This device controls a full HVAC system.
HVACSingle	This device only controls a single setpoint HVAC system.
Camera	This device will show a video stream.
Remote	This device brings up a Custom Remote Page.
Audio	This device controls a distributed audio system.
Scene4	This device brings up a four button scene panel.
Event	This device brings up controls to control an event.
Unknown	

## Appendix A.3.4. HVAC Modes

Mode	Description
HEAT	The device is in heat mode.
COOL	The device is in AC mode.
AUTO	The device will automatically switch between heat and AC.
OFF	The device is turned off.