EAGLE™

Energy Access Gateway Link to Ethernet

REST API Manual

Trademarks
Third-party brands and company Names mentioned herein may be trademarks and/or registered trademarks of their respective companies and are the sole property of their respective manufacturers.

Notice
The author(s) assumes no responsibility for any errors or omissions that may appear in this document nor does it make a commitment to update the information contained herein.

EAGLE™ – Energy Access Gateway Link to Ethernet

RFA-Z109
Version 1.0

REST API

Table of Contents

OVERVIEW

EAGLE™ REST Interface

The **EAGLE™** is an Ethernet device that communicates with a smart meter over a secured ZigBee wireless network. The **EAGLE™** is an endpoint on the ZigBee network that is authorized and authenticated to communicate with the smart meter. The **EAGLE™** is also an endpoint on an Ethernet network that uses HTTP (Hypertext Transfer Protocol) over TCP/IP (Transmission Control Protocol/Internet Protocol) to communicate with web-enabled entities. The REST (Representational State Transfer) API (Application Programming Interface) described in this document provides a mechanism to allow external applications to "pull" smart meter data from the **EAGLE™**.

Communications Model

The **EAGLE™** is a web-enabled device that sits on an Ethernet network and communicates with an external Relay Server using a VPN connection which is initiated by the **EAGLE™**. Applications can connect to the Relay Server using the REST API, and the Relay Server will route commands from the Application to the **EAGLE™**, and will route notifications from the **EAGLE™** to the Application.
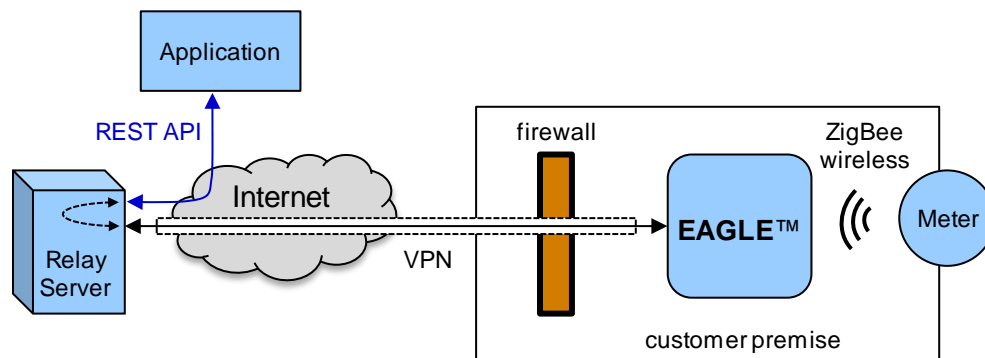
Applications that connect in this way can effectively communicate with the **EAGLE™** using 8-bit Extended ASCII characters (code page 1252) formed into HTTP POST requests. The body of these POST requests contains XML (eXtensible Markup Language) fragments. These XML fragments contain Commands for the **EAGLE™**.

The Relay Server responds to POSTs from the Application with valid HTTP responses. The body of these responses contains XML fragments that make up Notifications from **EAGLE™**, which are in response to the Commands sent by the Application.

The **EAGLE™** is identified by using its Cloud ID, which is made up of the last 6 digits of the Ethernet MACID that is shown on the label on the underside of the device. In order to be accessed remotely through the Relay Server, the **EAGLE™** must be registered with a Cloud Account on rainforestcloud.com. Instructions for this registration can be found in the *EAGLE™ User Manual, version 2.20*.

The registration process will associate a user email address and password with the **EAGLE**™. These will be needed to access the **EAGLE**™ through the Relay Server.

The Relay Server connection has the following characteristics:
- Host: rainforestcloud.com
- Port: 9445
- Protocol: HTTPS
- URL: /cgi-bin/post_manager

The **EAGLE**™ is also a ZigBee device. It is ZigBee Smart Energy 1.x compliant, and is certified by the ZigBee Alliance to operate according to that standard. Therefore, the API options are restricted to what is allowed by the ZigBee Smart Energy 1.x standard.

The **EAGLE**™ must be authorized and authenticated by the ZigBee Coordinator before the **EAGLE**™ can communicate with the smart meter. Generally, the smart meter is also the ZigBee Coordinator. In this case, the **EAGLE**™ is authorized by the owner of the smart meter (i.e. the electric utility); the owner needs the MAC ID and Install Code for the **EAGLE**™ in order to set up the authorization.

Data Structures

1. Commands

The Application sends Commands in HTTP POST requests. POST requests have the following structure:

```
POST <URL> HTTP/1.0
<headers>
<blank>
<body>
```

Where:
- Every line ends with the carriage return and line feed characters (0x0D, 0x0A).

- `<URL>` is the Uniform Resource Locator (web address) of the Relay Server.

- `<headers>` are a variable number of HTTP headers; each header is on its own line.
  The following items must be included in the headers:
    - **`Cloud-ID: xxxxxx`**
      where "`xxxxxx`" is the 6-digit Cloud ID for the target **EAGLE**™.
    - **`User: user_name`**
      where "`user_name`" is the user email address registered with this **EAGLE**™ on rainforestcloud.com.
    - **`Password: user_password`**
      where "`user_password`" is the user password registered with this **EAGLE**™ on rainforestcloud.com.
- `<blank>` is a blank line, consisting only of the carriage return and line feed characters (0x0D, 0x0A).

- `<body>` is the main text of the POST request, which has the structure shown below.

The body of the POST consists of XML Fragments. An XML Fragment is a stripped down XML Element. The **EAGLE™** uses XML Fragments to simplify the parsing of the data stream, while providing a data structure that is flexible and human readable.

The XML Fragments have the following structure:

```
<tag>
     <element>value</element>
     …
</tag>
```

Where:
- Every line ends with the carriage return and line feed characters (0x0D, 0x0A).
- `<tag>` is the start tag for the XML Fragment; each type will have a unique tag name;
- `<element>` is the start tag for an element; there will be one or more child elements in the fragment; each element will have a unique element name.
- … indicates the variable number of specific elements

Element values can be of various types:
- `{string}` indicates an element consisting of Extended ASCII text
- `{enumeration}` indicates an element that can have a specific list of values.
- `0xFFFFF` indicates an element consisting of a base16 (hex) number
- `00` indicates an element consisting of an integer
- `000.000` indicates an element consisting of a signed decimal number

`[<element>]` – square brackets indicate optional elements.
`value1|value2|value3` – vertical bars separate valid values in an enumeration list.

Note that element names are case insensitive; the case is used strictly for legibility. The **EAGLE™** will ignore case when receiving POST requests.

Example

Here is an example of a POST request:

```
POST /sample.php HTTP/1.0
Host: 192.168.11.3:8888
Accept: */*
Content-Length: 55
Content-Type: text/xml
Cloud-Id: 00ce69
Password: my123password
User: me@my_email.com

<LocalCommand>
<Name>get_instantaneous_demand</Name>
<MacId>0x00178d0000000004</MacId>
</LocalCommand>
```

Note that every line in the above example actually ends with the carriage return and line feed characters (0x0D, 0x0A). These are not shown explicitly for clarity.

2. Notifications

The **EAGLE™** generates a valid HTTP response to each POST request. These look like:

```
HTTP/1.0 <code>

<headers>

<blank>

<body>
```

Where:
- Every line ends with the carriage return and line feed characters (0x0D, 0x0A).
- `<code>` is an HTTP status code, which consists of a 3-digit number and a short text phrase. This is usually "200 OK".
- `<headers>` are a variable number of HTTP headers; each header is on its own line.
- `<blank>` is a blank line, consisting only of the carriage return and line feed characters (0x0D, 0x0A).
- `<body>` is the main text of the response, which contains the Notification, which has the structure shown below.
-

Example

Here is an example of a reply:

```
HTTP/1.0 200 OK
Date: Wed, 18 Dec 2013 21:28:44 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.3
Content-Length: 379
Connection: close
Content-Type: text/html; charset=UTF-8

<InstantaneousDemand>
<DeviceMacId>0x00158d0000000004</DeviceMacId>
<MeterMacId>0x00178d0000000004</MeterMacId>
<TimeStamp>0x185adc1d</TimeStamp>
<Demand>0x001738</Demand>
<Multiplier>0x00000001</Multiplier>
<Divisor>0x000003e8</Divisor>
<DigitsRight>0x03</DigitsRight>
<DigitsLeft>0x00</DigitsLeft>
<SuppressLeadingZero>Y</SuppressLeadingZero>
</InstantaneousDemand>
```

Note that every line in the above example actually ends with the carriage return and line feed characters (0x0D, 0x0A). These are not shown explicitly for clarity.

3. HTTP 1.1

An HTTP POST can also specify that HTTP 1.1 be used. In this case, the reply generated by the **EAGLE**™ will contain the header element "Transfer-Encoding: chunked" in place of the "Content-Length" header element. This indicates that the body of the response will be divided into chunks. Each chunk of data will be preceded by the chunk length on a line by itself as a hexadecimal number. Each chunk will also be followed by a blank line. The end of the message will be indicated by a zero on a line by itself.

4. JSON

Any of the Commands described in this document can include an optional XML tag: <Format>JSON</Format>. This will cause the Notification generated by the **EAGLE**™ to be in JSON (JavaScript Object Notation) format. If no Format tag is found, the **EAGLE**™ will use the default XML format.

Example

```
POST /sample/post_manager HTTP/1.1
Host: server.rainforestautomation.com
Accept: */*
Content-Type: text/xml
Content-Length: 80
Cloud-Id: 00038b
Password: 123asdf
User: you@your_email.com
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache

<Command>
  <Name>get_network_info</Name>
  <Format>JSON</Format>
</Command>


===== RESPONSE =====

HTTP/1.1 200 OK
Content-type: text/html
Transfer-Encoding: chunked
Date: Wed, 09 Jul 2014 22:30:23 GMT
Server: lighttpd/1.4.35

185
{ "Notification"="NetworkInfo",
  "DeviceMacId"="0x00158d00001aba29",
  "InstallCode"="0x91b98cd9c1d8aa9b",
  "LinkKeyHigh"="0xba8a14c201230175",
  "LinkKeyLow"="0x893644f99acc7c45",
  "FWVersion"="1.4.45 (6564)",
  "HWVersion"="3.2.1",
  "Manufacturer"="Rainforest Automation, Inc.",
  "ModelId"="Z109-EAGLE",
  "DateCode"="20120923ABCDEFGH",
  "ImageType"="0x1001",
  "Protocol"="Zigbee"}

0
```

COMMANDS & NOTIFICATIONS

1. Command: get_network_info

Send the **get_network_info** command to get information about a network interface on the **EAGLE™**. The **EAGLE™** will send a *NetworkInfo* notification in response.

```
<Command>
        <Name>get_network_info</Name>
        [<Protocol>{enumeration}</Protocol>]
        [<MacId>0xFFFFFFFFFFFFFFFF</MacId>]
</Command>
```

Element	Range	Description
Protocol	ZigBee; Optional	Type of network interface (only ZigBee supported at this time); if not specified, the default is ZigBee
MacId	16 hex digits; Optional	MAC Address of ZigBee radio, if Protocol=ZigBee

2. Notification: *NetworkInfo*

NetworkInfo notifications provide basic information about an **EAGLE™** network interface.

```
<NetworkInfo>
        <Protocol>{enumeration}</Protocol>
                        :
                protocol specific elements
                        :
</NetworkInfo>
```

Element	Range	Description
Protocol	ZigBee	Type of network interface (only ZigBee supported at this time)

2a. *NetworkInfo* (Protocol = ZigBee)

```
<NetworkInfo>
        <DeviceMacId>0xFFFFFFFFFFFFFFFF</DeviceMacId>
        <InstallCode>0xFFFFFFFFFFFFFFFF</InstallCode>
        <LinkKeyHigh>0xFFFFFFFFFFFFFFFF</LinkKeyHigh>
        <LinkKeyLow>0xFFFFFFFFFFFFFFFF</LinkKeyLow>
        <FWVersion>{string}</FWVersion>
        <HWVersion>{string}</HWVersion>
        <Manufacturer>{string}</Manufacturer>
        <ModelId>{string}</ModelId>
        <DateCode>{string}</DateCode>
        <ImageType>0xFFFF</ImageType>
        <Protocol>ZigBee</Protocol>
</NetworkInfo>
```

Element	Range	Description
DeviceMacId	16 hex digits	MAC Address of **EAGLE**™ ZigBee radio
InstallCode	16 hex digits	Install Code for **EAGLE**™ ZigBee radio
LinkKeyHigh	16 hex digits	Upper 16 digits of ZigBee radio Link Key
LinkKeyLow	16 hex digits	Lower 16 digits of ZigBee radio Link Key
FWVersion	Text	Firmware Version
HWVersion	Text	Hardware Version
Manufacturer	Text	"Rainforest Automation"
ModelId	Text	"RFA-Z109"
DateCode	YYYYMMDDZZZZZZZZ	Manufacturer's date code and lot number
ImageType	4 hex digits	ZigBee code image type

3. Command: get_network_status

Send the **get_network_status** command to get detailed information about the status of a network that the **EAGLE**™ is connected to. In response, the **EAGLE**™ will send a *NetworkStatus* notification.

```
<Command>
        <Name>get_network_status</Name>
        [<Protocol>{enumeration}</Protocol>]
        [<MacId>0xFFFFFFFFFFFFFFFF</MacId>]
</Command>
```

Element	Range	Description
Protocol	ZigBee; Optional	Type of network interface (only ZigBee supported at this time); if not specified, the default is ZigBee
MacId	16 hex digits; Optional	MAC Address of ZigBee radio, if Protocol=ZigBee

4. Notification: *NetworkStatus*

NetworkStatus notifications provide detailed information about a network that the **EAGLE**™ is connected to.

```
<NetworkStatus>
        <Protocol>{enumeration}</Protocol>
                        :
             protocol specific elements
                        :
</NetworkStatus>
```

Element	Range	Description
Protocol	ZigBee	Type of network interface (only ZigBee supported at this time)

4a. *NetworkStatus* (Protocol = ZigBee)

```
<NetworkStatus>
      <Protocol>ZigBee</Protocol>
      <DeviceMacId>0xFFFFFFFFFFFFFFFF</DeviceMacId>
      <Status>{enumeration}</Status>
      <CoordMacId>0xFFFFFFFFFFFFFFFF</CoordMacId>
      [<Description>{string}</Description>]
      [<StatusCode>0xFF</StatusCode>]
      [<ExtPanId>0xFFFFFFFFFFFFFFFF</ExtPanId>]
      [<ShortAddr>0xFFFF</ShortAddr>]
      [<Channel>00</Channel>]
      <LinkStrength>0xFF</LinkStrength>
</NetworkStatus>
```

Element	Range	Description
DeviceMacId	16 hex digits	MAC Address of **EAGLE™** ZigBee radio
Status	Initializing | Network Discovery | Joining | Join: Fail | Join: Success | Authenticating | Authenticating: Success | Authenticating: Fail | Connected | Disconnected | Rejoining	Indicates the current state of the **EAGLE™** ZigBee radio.
CoordMacId	16 hex digits	MAC Address of Meter
Description	Text; Optional	Description of ZigBee radio.state
StatusCode	2 hex digits; Optional	Status code for the current state
ExtPanId	16 hex digits; Optional	Extended PAN ID of the ZigBee network
ShortAddr	4 hex digits; Optional	The short address assigned to the **EAGLE™** by the network coordinator
Channel	11 – 26; Optional	Indicates the radio channel on which the **EAGLE™** is operating
LinkStrength	0x00 – 0x64	Indicates the strength of the radio link

5. Command: get_instantaneous_demand

Send the **get_ instantaneous _demand** command to get the real time demand from the meter. The **EAGLE™** will send an *InstantaneousDemand* notification in response.

```
<Command>
      <Name>get_instantaneous_demand</Name>
      <MacId>0xFFFFFFFFFFFFFFFF</MacId>
</Command>
```

Element	Range	Description
MacId	16 hex digits	MAC Address of **EAGLE™** ZigBee radio

6. Notify: *InstantaneousDemand*

InstantaneousDemand notification provides the current consumption rate as recorded by the meter. Note that readings are recorded as integers and are converted into a decimal number by using the multiplier and divisor, i.e., Demand Reading = demand value X multiplier / divisor.

```
<InstantaneousDemand>
      <DeviceMacId>0xFFFFFFFFFFFFFFFF</DeviceMacId>
      <MeterMacId>0xFFFFFFFFFFFFFFFF</MeterMacId>
      <TimeStamp>0xFFFFFFFF</TimeStamp>
      <Demand>0xFFFFFF</Demand>
      <Multiplier>0xFFFFFFFF</Multiplier>
      <Divisor>0xFFFFFFFF</Divisor>
      <DigitsRight>0xFF</DigitsRight>
      <DigitsLeft>0xFF</DigitsLeft>
      <SuppressLeadingZero>{enumeration}</SuppressLeadingZero>
</InstantaneousDemand>
```

Element	Range	Description
DeviceMacId	16 hex digits	MAC Address of **EAGLE™** ZigBee radio
MeterMacId	16 hex digits	MAC Address of Meter
TimeStamp	Up to 8 hex digits	UTC Time (offset in seconds from 00:00:00 01Jan2000) when demand data was received from meter.
Demand	Up to 6 hex digits	The raw instantaneous demand value
Multiplier	Up to 8 hex digits	The multiplier; if zero, use 1
Divisor	Up to 8 hex digits	The divisor; if zero, use 1
DigitsRight	Up to 2 hex digits	Number of digits to the right of the decimal point to display
DigitsLeft	Up to 2 hex digits	Number of digits to the left of the decimal point to display

Suppress LeadingZero	Y | N	Y: Do not display leading zeros N: Display leading zeros

7. Command: get_price

Send the **get_price** command to get price information from the meter. The **EAGLE**™ will send an *PriceCluster* notification in response

```
<Command>
      <Name>get_price</Name>
      <MacId>0xFFFFFFFFFFFFFFFF</MacId>
</Command>
```

Element	Range	Description
MacId	16 hex digits	MAC Address of **EAGLE**™ ZigBee radio

8. Notify: *PriceCluster*

PriceCluster notification provides the current price in effect in the meter. If the meter price is not set, then the price returned is zero.

```
<PriceCluster>
      <DeviceMacId>0xFFFFFFFFFFFFFFFF</DeviceMacId>
      <MeterMacId>0xFFFFFFFFFFFFFFFF</MeterMacId>
      <TimeStamp>0xFFFFFFFF</TimeStamp>
      <Price>0xFFFFFFFF</Price>
      <Currency>0xFFFF</Currency>
      <TrailingDigits>0xFF</TrailingDigits>
      <Tier>00</Tier>
      [<TierLabel>{string}</TierLabel> |
      <RateLabel>{string}</RateLabel>]
</PriceCluster>
```

Element	Range	Description
DeviceMacId	16 hex digits	MAC Address of **EAGLE**™ ZigBee radio
MeterMacId	16 hex digits	MAC Address of Meter
TimeStamp	Up to 8 hex digits	UTC Time (offset in seconds from 00:00:00 01Jan2000) when price data was received from meter or set by user
Price	Up to 8 hex digits	Price from meter or set by user; will be zero if no price is set
Currency	Up to 4 hex digits	Currency being used; value of this field matches the values defined by ISO 4217

TrailingDigits	Up to 2 hex digits	The number of implicit decimal places in the price. (e.g. 2 means divide Price by 100).
Tier	1 – 5	The price Tier in effect.
RateLabel	Text; Optional	Rate label for the current price tier; will be "Set by User" if a user-defined price is set

9. Command: get_message

Send the **get_message** command to get the current text message from the meter. The **EAGLE™** will send a *MessageCluster* notification in response.

```
<Command>
      <Name>get_message</Name>
      <MacId>0xFFFFFFFFFFFFFFFF</MacId>
</Command>
```

Element	Range	Description
MacId	16 hex digits	MAC Address of **EAGLE™** ZigBee radio

10. Notify: *MessageCluster*

MessageCluster notifications provide the current text message from the meter. If a confirmation is required, the ConfirmationRequired flag is set. If the user has already confirmed the message, then the Confirmed flag is set to Y. The ID is the reference to a particular message. The message text is HTML escape encoded.

```
<MessageCluster>
      <DeviceMacId>0xFFFFFFFFFFFFFFFF</DeviceMacId>
      <MeterMacId>0xFFFFFFFFFFFFFFFF</MeterMacId>
      <TimeStamp>0xFFFFFFFF</TimeStamp>
      <Id>0xFFFFFFFF</Id>
      <Priority>{enumeration}</Priority>
      <Text>{string}</Text>
      <ConfirmationRequired>{enumeration}</ConfirmationRequired>
      <Confirmed>{enumeration}</Confirmed>
      <Read>{enumeration}</Read>
      <Queue>{enumeration}</Queue>
</MessageCluster>
```

Element	Range	Description
DeviceMacId	16 hex digits	MAC Address of **EAGLE™** ZigBee radio
MeterMacId	16 hex digits	MAC Address of Meter
TimeStamp	Up to 8 hex digits	UTC Time (offset in seconds from 00:00:00 01Jan2000) when message was received from meter

Id	Up to 8 hex digits	Message ID from meter
Priority	Low | Medium | High | Critical	Message priority
Text	Text	Contents of message, HTML encoded: > replaces the > character < replaces the < character & replaces the & character " replaces the " character
Confirmation Required	Y | N	Y: a user confirmation is required; N: a user confirmation is not required (default)
Confirmed	Y | N	Y: the user confirmation has been sent; N: the user confirmation has not been sent (default)
Read	Y | N	Y: message has been viewed by user; N: message has not been viewed by user
Queue	Active | Cancel Pending	Active: Indicates message is in active queue Cancel Pending: Indicates message is in cancel pending queue

11. Command: confirm_message

Send the **confirm_message** command to have the **EAGLE**™ confirm the message as indicated by the ID. To verify that the message confirmation was sent, use a **get_message** command. The resulting *MessageCluster* notification should show Confirmed=Y.

```
<Command>
     <Name>confirm_message</Name>
     <MacId>0xFFFFFFFFFFFFFFFF</MacId>
     <Id>0xFFFFFFFF</Id>
</Command>
```

Element	Range	Description
MacId	16 hex digits	MAC Address of **EAGLE**™ ZigBee radio
ID	0x0 – 0xFFFFFFFF	Message ID to confirm

12. Command: get_current_summation

Send the **get_current_summation** command to get the total consumption to date as recorded by the meter. The **EAGLE**™ will send a *CurrentSummation* notification in response.

```
<Command>
     <Name>get_current_summation</Name>
     <MacId>0xFFFFFFFFFFFFFFFF</MacId>
</Command>
```

Element	Range	Description
MacId	16 hex digits	MAC Address of **EAGLE**™ ZigBee radio

13. Notify: *CurrentSummation*

CurrentSummation notification provides the total consumption to date as recorded by the meter. Note that readings are recorded as integers and are converted into a decimal number by using the multiplier and divisor, i.e., Summation Reading = summation value X multiplier / divisor.

```
<CurrentSummation>
      <DeviceMacId>0xFFFFFFFFFFFFFFFF</DeviceMacId>
      <MeterMacId>0xFFFFFFFFFFFFFFFF</MeterMacId>
      <TimeStamp>0xFFFFFFFF</TimeStamp>
      <SummationDelivered>0xFFFFFFFF</SummationDelivered>
      <SummationReceived>0xFFFFFFFF</SummationReceived>
      <Multiplier>0xFFFFFFFF</Multiplier>
      <Divisor>0xFFFFFFFF</Divisor>
      <DigitsRight>0xFF</DigitsRight>
      <DigitsLeft>0xFF</DigitsLeft>
      <SuppressLeadingZero>{enumeration}</SuppressLeadingZero>
</CurrentSummation>
```

Element	Range	Description
DeviceMacId	16 hex digits	MAC Address of **EAGLE**™ ZigBee radio
MeterMacId	16 hex digits	MAC Address of Meter
TimeStamp	Up to 8 hex digits	UTC Time (offset in seconds from 00:00:00 01Jan2000) when data was received from meter.
Summation Delivered	Up to 8 hex digits	The raw value of the total summation of commodity delivered from the utility to the user.
Summation Received	Up to 8 hex digits	The raw value of the total summation of commodity received from the user by the utility.
Multiplier	Up to 8 hex digits	The multiplier; if zero, use 1
Divisor	Up to 8 hex digits	The divisor; if zero, use 1
DigitsRight	Up to 2 hex digits	Number of digits to the right of the decimal point to display
DigitsLeft	Up to 2 hex digits	Number of digits to the left of the decimal point to display
Suppress LeadingZero	Y | N	Y: Do not display leading zeros N: Display leading zeros

14. Command: get_history_data

Send the **get_history_data** command to get a series of summation values over an interval of time. In response, the **EAGLE™** will send a *HistoryData* notification containing a series of *CurrentSummation* fragments – one for each sample over the interval.

```
<Command>
      <Name>get_history_data</Name>
      <MacId>0xFFFFFFFFFFFFFFFF</MacId>
      <StartTime>0xFFFFFFFF</StartTime>
      [<EndTime>0xFFFFFFFF</EndTime>]
      [<Frequency>0x0000</Frequency>]
</Command>
```

Element	Range	Description
MacId	16 hex digits	MAC Address of **EAGLE™** ZigBee radio
StartTime	Up to 8 hex digits	UTC Time (offset in seconds from 00:00:00 01Jan2000) of the start of the history interval. If StartTime is older than the oldest sample, then the interval will start from the oldest sample.
EndTime	Up to 8 hex digits; Optional	UTC Time (offset in seconds from 00:00:00 01Jan2000) of the end of the history interval. If not specified, EndTime will be the current time.
Frequency	4 hex digits; Optional	Requested number of seconds between samples. Availability limited by what has been collected by **EAGLE™**. Default is all of the existing samples in the interval.

15. Notify: *HistoryData*

A *HistoryData* notification contains a series of elements; each element is a *CurrentSummation* fragment. It is sent in response to a **get_history_data** command.

```
<HistoryData>
  <CurrentSummation>
      <DeviceMacId>0xFFFFFFFFFFFFFFFF</DeviceMacId>
      <MeterMacId>0xFFFFFFFFFFFFFFFF</MeterMacId>
      <TimeStamp>0xFFFFFFFF</TimeStamp>
      <SummationDelivered>0xFFFFFFFF</SummationDelivered>
      <SummationReceived>0xFFFFFFFF</SummationReceived>
      <Multiplier>0xFFFFFFFF</Multiplier>
      <Divisor>0xFFFFFFFF</Divisor>
      <DigitsRight>0xFF</DigitsRight>
      <DigitsLeft>0xFF</DigitsLeft>
      <SuppressLeadingZero>{enumeration}</SuppressLeadingZero>
  </CurrentSummation>
            :
```

```
                :
        <CurrentSummation>
            <DeviceMacId>0xFFFFFFFFFFFFFFFF</DeviceMacId>
            <MeterMacId>0xFFFFFFFFFFFFFFFF</MeterMacId>
            <TimeStamp>0xFFFFFFFF</TimeStamp>
            <SummationDelivered>0xFFFFFFFF</SummationDelivered>
            <SummationReceived>0xFFFFFFFF</SummationReceived>
            <Multiplier>0xFFFFFFFF</Multiplier>
            <Divisor>0xFFFFFFFF</Divisor>
            <DigitsRight>0xFF</DigitsRight>
            <DigitsLeft>0xFF</DigitsLeft>
            <SuppressLeadingZero>{enumeration}</SuppressLeadingZero>
        </CurrentSummation>
    </HistoryData>
```

16. Command: set_schedule

Send the **set_schedule** command to change how the **EAGLE™** polls the meter. The rate at which each type of meter reading is polled can be set.

```
    <Command>
        <Name>set_schedule</Name>
        <DeviceMacId>0xFFFFFFFFFFFFFFFF</DeviceMacId>
        <Event>{enumeration}</Event>
        <Frequency>0xffff</Frequency>
        <Enabled>{enumeration}</Enabled>
    </Command>
```

Element	Range	Description
MacId	16 hex digits	MAC Address of **EAGLE™** ZigBee radio
Event	time | message | price | summation | demand | scheduled_prices | profile_data | billing_period | block_period |	Type of meter reading to schedule
Frequency	4 hex digits	Frequency to poll meter, in seconds
Enabled	Y | N	Y: Set this rate for this type of reading N: Disable polling for this type of reading

17. Command: get_schedule

Send the **get_schedule** command to get the **EAGLE™** scheduler information. The **EAGLE™** will send the *ScheduleInfo* notification in response; or, the **EAGLE™** will send a series of *ScheduleInfo* notifications if the Event field is omitted.

```
<Command>
      <Name>get_schedule</Name>
      <MeterMacId>0xFFFFFFFFFFFFFFFF</MeterMacId>
      [<Event>{enumeration}</Event>]
</Command>
```

Element	Range	Description
DeviceMacId	16 hex digits	MAC Address of **EAGLE™** ZigBee radio
Event	time | message | price | summation | demand | scheduled_prices | profile_data | billing_period | block_period |	Type of meter reading to get schedule info for. If this is omitted, then schedule info for all events is requested.

18. Notify: *ScheduleInfo*

A *ScheduleInfo* notification provides the frequency at which a certain event is scheduled to happen, and if it is enabled.

```
<ScheduleInfo>
      <DeviceMacId>0xFFFFFFFFFFFFFFFF</DeviceMacId>
      <MeterMacId>0xFFFFFFFFFFFFFFFF</MeterMacId>
      <Event>{enumeration}</Event>
      <Frequency>0x00000000</Frequency>
      <Enabled>{enumeration}</Enabled>
</ScheduleInfo>
```

Element	Range	Description
DeviceMacId	16 hex digits	MAC Address of **EAGLE™** ZigBee radio
MeterMacId	16 hex digits	Unique MAC Address of meter
Event	time | message | price | summation | demand | scheduled_prices | profile_data | billing_period | block_period |	Type of event scheduled
Frequency	4 hex digits	The frequency in seconds the event will be executed
Enabled	Y | N	Y: the scheduled event will execute; N: the scheduled event will not execute.

19. Command: reboot

Send the **reboot** command to get the **EAGLE™** to restart the specified part of its system.

```
<Command>
    <Name>reboot</Name>
    <MacId>0xFFFFFFFFFFFFFFFF</MacId>
    <Target>{enumeration}</Target>
</Command>
```

Element	Range	Description
MacId	16 hex digits	MAC Address of **EAGLE™** ZigBee radio
Target	Zigbee | Eagle | All	Part of the device to be restarted: Zigbee – ZigBee radio Eagle – Linux system All – every part of the device

HANDLING THE NUMBERS

Summation and Demand Readings

Summation and Demand readings from the meter are reported as hexadecimal integers with formatting instructions to calculate the decimal value.

Here is a typical *InstantaneousDemand* notification:

```
<InstantaneousDemand>
<DeviceMacId>0x00158d0000000004</DeviceMacId>
<MeterMacId>0x00178d0000000004</MeterMacId>
<TimeStamp>0x185adc1d</TimeStamp>
<Demand>0x001738</Demand>
<Multiplier>0x00000001</Multiplier>
<Divisor>0x000003e8</Divisor>
<DigitsRight>0x03</DigitsRight>
<DigitsLeft>0x00</DigitsLeft>
<SuppressLeadingZero>Y</SuppressLeadingZero>
</InstantaneousDemand>
```

The values from this notification are:

- Demand reading = 1738 (hex) = 5944 (decimal)
- Multiplier = 1
- Divisor = 3E8 (hex) = 1000 (decimal)

The actual Demand value is calculated by using the multiplier and divisor:

- 5944 x 1 / 1000 = 5.944

If the multiplier or divisor is zero then use a value of one instead.

For Summation, the same process is used, and the multiplier and divisor are used by both the SummationDelivered and SummationReceived readings in the *CurrentSummation* notification.

The other formatting values (DigitsRight, DigitsLeft, SupressLeadingZeros) are the preferred display settings set by the utility and can be ignored for the purposes of calculating values.

Price Readings

Price readings from the meter are reported as a hexadecimal integer with a format instruction to place the decimal point.

Here is a typical *PriceCluster* notification:

```
<PriceCluster>
  <DeviceMacId>0x00158d00001ab01e</DeviceMacId>
  <MeterMacId>0x00078100011cf431</MeterMacId>
  <TimeStamp>0x1a462b4d</TimeStamp>
  <Price>0x0000007d</Price>
  <Currency>0x0348</Currency>
  <TrailingDigits>0x03</TrailingDigits>
  <Tier>0x01</Tier>
```

```
        <RateLabel>Set by User</RateLabel>
    </PriceCluster>
```

The values from this notification are:

- Price reading = 7D (hex) = 125 (decimal)
- Trailing Digits = 3
- Currency Code = 348 (hex) = 840 (decimal)

The actual value of the Price is determined by placing the decimal point 3 digits from the right. The Currency Code indicates that the value is in US Dollars (see ISO 4217). So the final price is $0.125.