rainforest
a u t o m a t i o n

EAGLE™

Energy Access Gateway Link to Ethernet

Uploader API Manual

Version 6
July 2014

EAGLE™ – Energy Access Gateway Link to Ethernet

RFA-Z109
Version 6

Uploader API Manual

Table of Contents

OVERVIEW

EAGLE™ Uploader API

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 API (Application Programming Interface) described in this document provides a mechanism to allow external web servers to receive smart meter data from the **EAGLE**™.

Communications Model

The **EAGLE**™ is a Linux-based device that is running an internal web server. This web server receives real time data from the smart meter, and can forward this data to external web servers using HTTP POST requests. The body of these POST requests contains the meter data structured as XML Fragments.

Data Structures

1. Requests

The **EAGLE**™ sends data in a POST request. POST requests have the following structure:

```
POST <URL> HTTP/1.1
<headers>
<blank>
<body>
```

Where:
- Every line ends with the newline character (0x0A).
- `<URL>` is the Uniform Resource Locator (web address) of the external web server.
- `<headers>` are a variable number of HTTP headers; each header is on its own line.
- `<blank>` is a blank line, consisting only of the newline character (0x0A).
- `<body>` is the main text of the POST request, which has the structure shown below.

The body of the POST request has the following structure:

```
<?xml version="1.0"?>
<rainforest macId="0xffffffffffff" version="undefined"
timestamp="0000000000s">
<XML>
</rainforest>
```

The first line is the standard XML (eXtensible Markup Language) Prolog. This is followed by the Root Element, which for our purposes is enclosed by the tags `<rainforest>` and `</rainforest>`. The Root Element has three attributes:

- macId – this is a 12-digit hexadecimal number. It is the MAC Address of the **EAGLE**™.
- version – this is for future use and is currently undefined.
- timestamp -- this is an integer with a standard Unix timestamp, i.e., number of seconds since Jan.1, 1970.

The body of the Root Element 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 newline character (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. XML parsers should be designed to ignore case when receiving requests from the **EAGLE**™.

Example

Here is an example of a POST request:

```
POST /rfaeagle.php HTTP/1.0
Host: 192.168.11.3:8888
Accept: */*
From: nobody@rainforestautomation.com
User-Agent: Raven Uploader/v1
Content-Length: 483
Content-Type: application/x-www-form-urlencoded

<?xml version="1.0"?>
<rainforest macId="0xf0ad4e00ce69" timestamp="1355292588s">
```

```
<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>
</rainForest>
```

Note that every line in the above example actually ends with the newline character (0x0A). This is not shown explicitly for clarity.

2. Replies

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

```
HTTP/1.0 <code>

<headers>

<blank>

<body>
```

Where:
- Every line ends with the newline character (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 newline character (0x0A).
- `<body>` is optional and can contain a single XML fragment.

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: 168
Connection: close
Content-Type: text/html; charset=UTF-8

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

Rainforest Automation, Inc.

```
<StartTime>0x0</StartTime>
<Demand>0x001738</Demand>
<Frequency>300</Frequency>
</LocalCommand>
```

Note that every line in the above example actually ends with the newline character (0x0A). This is not shown explicitly for clarity.

Most replies will not have a `<body>` component, and will simply acknowledge a POST sent by the **EAGLE™**. However, it is also possible to send a single XML-formatted command to the **EAGLE™** in the `<body>` of the reply. While a command sent in this way is being processed, the **EAGLE™** will not accept any additional commands found in a reply. So, only one command can be put in the `<body>` of a reply.

A command sent to the **EAGLE™** in the `<body>` of a reply may change the operation of the gateway, and may also cause the generation of a POST request containing specific data.

XML FRAGMENTS

1. DeviceInfo

DeviceInfo fragments provide some basic information about the **EAGLE™** device.

```
<DeviceInfo>
    <DeviceMacId>0xFFFFFFFFFFFFFFFF</DeviceMacId>
    <InstallCode>0xFFFFFFFFFFFFFFFF</InstallCode>
    <LinkKey>0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF</LinkKey>
    <FWVersion>{string}</FWVersion>
    <HWVersion>{string}</HWVersion>
    <ImageType>0xFFFF</ImageType>
    <Manufacturer>{string}</Manufacturer>
    <ModelId>{string}</ModelId>
    <DateCode>{string}</DateCode>
</DeviceInfo>
```

Element	Range	Description
DeviceMacId	16 hex digits	MAC Address of **EAGLE™** ZigBee radio
InstallCode	16 hex digits	Install Code for **EAGLE™** ZigBee radio
LinkKey	32 hex digits	ZigBee radio Link Key
FWVersion	Text	Firmware Version
HWVersion	Text	Hardware Version
ImageType	4 hex digits	ZigBee code image type
Manufacturer	Text	"Rainforest Automation"
ModelId	Text	"RFA-Z109"
DateCode	YYYYMMDDZZZZZZZZ	Manufacturer's date code and lot number

2. NetworkInfo

NetworkInfo fragments provide detailed information about the ZigBee network that the **EAGLE™** is connected to.

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

Element	Range	Description
DeviceMacId	16 hex digits	MAC Address of **EAGLE**™ ZigBee radio
CoordMacId	16 hex digits	MAC Address of Meter
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.
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
Channel	11 – 26; Optional	Indicates the radio channel on which the **EAGLE**™ is operating
ShortAddr	4 hex digits; Optional	The short address assigned to the **EAGLE**™ by the network coordinator
LinkStrength	0x00 – 0x64	Indicates the strength of the radio link

3. InstantaneousDemand

InstantaneousDemand fragments provide 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	6 hex digits	The raw instantaneous demand value. This is a 24-bit signed integer.
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

4. PriceCluster

PriceCluster fragments provide the current price in effect on the meter, or the user-defined price set on the **EAGLE™**. If the user-defined price is not set and 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	Rate label for the current price tier; will be "Set by User" if a user-defined price is set

5. Message

Message fragments 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>
      <Text>{string}</Text>
      <Priority>{enumeration}</Priority>
      <ConfirmationRequired>{enumeration}</ConfirmationRequired>
      <Confirmed>{enumeration}</Confirmed>
      <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
Text	Text	Contents of message, HTML encoded: > replaces the > character < replaces the < character & replaces the & character " replaces the " character
Priority	Low | Medium | High | Critical	Message priority
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)
Queue	Active | Cancel Pending	Active: Indicates message is in active queue Cancel Pending: Indicates message is in cancel pending queue

6. CurrentSummation

CurrentSummation fragments 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.

```
<CurrentSummationDelivered>
      <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>
</CurrentSummationDelivered>
```

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.
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

7. MeterInfo

MeterInfo fragments provide information about meters that are on the ZigBee network.

```
<MeterInfo>
    <DeviceMacId>0xFFFFFFFFFFFFFFFF</DeviceMacId>
    <MeterMacId>0xFFFFFFFFFFFFFFFF</MeterMacId>
    <Type>{enumeration}</Type>
    <NickName>{string}</NickName>
    [<Account>{string}</Account>]
    [<Auth>{string}</Auth>]
    [<Host>{string}</Host>]
    [<Enabled>{enumeration}</Enabled>]
</MeterInfo>
```

Element	Range	Description
DeviceMacId	16 hex digits	MAC Address of **EAGLE™** ZigBee radio
MeterMacId	16 hex digits	MAC Address of Meter
Type	electric | gas | water | other	Type of meter
Nickname	Text	Nickname set for the meter
Account	Text	Account Identification
Auth	Text	Authentication code
Host	Text	Hosting Provider
Enabled	Y | N	Y: to start transmitting data to host N: to stop transmitting data to host

8. FastPollStatus

FastPollStatus fragments provide the current status of fast poll mode, as negotiated with the meter. A value of zero for Duration indicates that fast poll mode is not active.

```
<FastPollStatus>
    <DeviceMacId>0xFFFFFFFFFFFFFFFF</DeviceMacId>
    <MeterMacId>0xFFFFFFFFFFFFFFFF</MeterMacId>
    <Frequency>0x00</Frequency>
    <EndTime>0xFFFFFFFF</EndTime>
</FastPollStatus>
```

Element	Range	Description
DeviceMacId	16 hex digits	MAC Address of **EAGLE™** ZigBee radio
MeterMacId	16 hex digits	MAC Address of Meter
Frequency	0x01 – 0xFF	Frequency to poll meter, in seconds

EndTime	Up to 8 hex digits	UTC Time (offset in seconds from 00:00:00 01Jan2000) when fast poll will end. If EndTime is earlier than the current time, then fast poll mode is turned off.

9. HistoryData

A *HistoryData* fragment 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>
```

10. ProfileData

A *ProfileData* fragment is sent in response to a **get_profile_data** command. It contains a set of interval data points that have been stored by the meter (support for this feature varies by meter and utility). Each data point represents the change in total summation reading over a specified time interval. This time interval is reported in the ProfileIntervalPeriod field. The type of interval data (summation delivered or received) depends upon the IntervalChannel field in the **get_profile_data** command. Data is organized in reverse chronological order: the most recent interval is transmitted first and the oldest interval is transmitted last.

```
<ProfileData>
     <DeviceMacId>0xFFFFFFFFFFFFFFFF</DeviceMacId>
     <MeterMacId>0xFFFFFFFFFFFFFFFF</MeterMacId>
     <EndTime>0x00000000</EndTime>
     <Status>0x00</Status>
     <ProfileIntervalPeriod>0</ProfileIntervalPeriod>
     <NumberOfPeriodsDelivered>0x00</NumberOfPeriodsDelivered>
     [<IntervalData1>0x000000</IntervalData1>]
     [<IntervalData2>0x000000</IntervalData2>]
                    ⋮
                    ⋮
     [<IntervalData12>0x000000</IntervalData12>]
</ProfileData>
```

Element	Range	Description
DeviceMacId	16 hex digits	MAC Address of **EAGLE**™ ZigBee radio
MeterMacId	16 hex digits	MAC Address of Meter
EndTime	Up to 8 hex digits	UTC Time (offset in seconds from 00:00:00 01Jan2000) of the end of the most chronologically recent interval; 0x0 indicates the most recent interval block.
Status	0x0 – 0x05	Status of returned data:

0x00	Success	
0x01	Undefined Interval Channel requested	
0x02	Interval Channel not supported	
0x03	Invalid End Time	
0x04	More periods Requested than can be returned	
0x05	No intervals available for the requested time	

Element	Range	Description
ProfileIntervalPeriod	0 - 7	The length of each sampling interval:

0	Daily
1	60 minutes
2	30 minutes
3	15 minutes
4	10 minutes
5	7.5 minutes
6	5 minutes
7	2.5 minutes

Element	Range	Description
NumberOfPeriodsDelivered	0x0 – 0xFF	The number of intervals being returned.
IntervalDataX	Up to 6 hex digits	X = 1-12. Series of up to 12 interval data points from the meter. Most recent interval is first; oldest is last. Invalid intervals are marked as 0xFFFFFF.

XML COMMANDS

These XML-formatted commands can be used in the body of a reply to change the operation of the **EAGLE™**, and to cause the **EAGLE™** to generate a POST request containing specific data. The `<body>` of a reply can only contain one command.

1. Command: set_fast_poll

Send the **set_fast_poll** command to have the **EAGLE™** set the fast poll mode on the meter. In fast poll mode, the meter will send Instantaneous Demand updates at the frequency requested. This is a ZigBee Smart Energy 1.1 feature.

For ZigBee Smart Energy 1.0 meters, the **EAGLE™** will emulate this feature, if possible. For some meters fast poll mode will not be allowed. In that case, polling will default to a maximum frequency of every 4 seconds for up to 15 minutes.

If Duration is zero, then fast poll mode is turned off.

Note that the tag <RavenCommand> is used instead of <LocalCommand>.

```
<RavenCommand>
    <Name>set_fast_poll</Name>
    <MacId>0xFFFFFFFFFFFFFFFF</MacId>
    <Frequency>0x00</Frequency>
    <Duration>0x00</Duration>
</RavenCommand>
```

Element	Range	Description
MacId	16 hex digits	MAC Address of **EAGLE™** ZigBee radio
Frequency	0x01 – 0xFF	Frequency to poll meter, in seconds
Duration	0x00 – 0x0F	Duration of fast poll mode, in minutes; maximum is 15 minutes

2. Command: get_ fast_poll_status

Send the **get_fast_poll_status** command to get the current status of fast poll mode. The **EAGLE™** will send a *FastPollStatus* fragment in response.

```
<LocalCommand>
    <Name>get_fast_poll_status</Name>
    <MacId>0xFFFFFFFFFFFFFFFF</MacId>
</LocalCommand>
```

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

3. 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* fragment containing a series of *CurrentSummation* fragments – one for each sample over the interval.

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

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	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	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.

4. Command: get_ profile_data

Send the **get_profile_data** command to have the **EAGLE™** retrieve interval data information stored in the meter. The **EAGLE™** will send a *ProfileData* fragment in response.

Note that the tag <RavenCommand> is used instead of <LocalCommand>.

```
<RavenCommand>
      <Name>get_profile_data</Name>
      <DeviceMacId>0xFFFFFFFFFFFFFFFF</DeviceMacId>
      [<MeterMacId>0xFFFFFFFFFFFFFFFF</MeterMacId>]
      <NumberOfPeriods>0x00</NumberOfPeriods>
      <EndTime>0xFFFFFFFF</EndTime>
      <IntervalChannel>{enumeration}</IntervalChannel>
</RavenCommand>
```

Element	Range	Description
DeviceMacId	16 hex digits	MAC Address of **EAGLE**™ ZigBee radio
MeterMacId	16 hex digits	Unique MAC Address of meter. Optional if there is only one meter.
NumberOfPeriods	0x0 – 0xc	Number of intervals requested; maximum is 12.
EndTime	Up to 8 hex digits	UTC Time (offset in seconds from 00:00:00 01Jan2000) of the end of the most chronologically recent interval; 0x0 indicates the most recent interval block.
IntervalChannel	Delivered | Received	Delivered: Interval data for commodity delivered by the utility to the user. Received: Interval data for commodity received by the utility from the user.

5. 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.

Note that the tag <RavenCommand> is used instead of <LocalCommand>.

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

Element	Range	Description
DeviceMacId	16 hex digits	MAC Address of **EAGLE**™ ZigBee radio
Event	price | summation | demand | message	Type of reading to get from meter
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

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 kWh

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.