



# Cloud Uploader Specification

Version 2.2  
September 27, 2017

Copyright © 2017 by RAINFOREST AUTOMATION, INC (“RFA”). All rights reserved.

No part of this manual may be reproduced or transmitted in any form without the expressed, written permission of RFA.

Under copyright law, this manual or the software described within, cannot be copied, in whole or part, without the written consent of the manufacturer, except in the normal use of the software to make a backup copy. The same proprietary and copyright notices must be affixed to any permitted copies as were affixed to the original. This exception does not allow copies to be made for others, whether or not sold, but all of the material purchased (with all backup copies) can be sold, given, or loaned to another person. Under the law, copying includes translating into another language or format.

Rainforest Automation may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Rainforest Automation, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

## **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.

## 1. Overview

Rainforest devices like the EAGLE gateway have the ability to “upload” meter and device data to a cloud server on the Internet. This document specifies the various configuration options available as well as the API that is used for the upload.

### 1.1. Configuration

#### 1.1.1. Modes of Operation

The Uploader has 2 modes of operation:

1. Streaming: Data is streamed by the gateway as it is received from the devices and is not stored. This can be enabled by setting UploadSize to 0.
2. Buffered upload: This is where data is uploaded only after a certain, configurable parameters have been reached. In this case data packets are collected until either a configurable number of packets has been received, or some timeout has been reached. Again, if the upload site is not available, data packets will be stored within the Rainforest device until the upload site can be reached again.

#### 1.1.2. Configuration Files

There is a configuration file for each known upload site. The files are in XML format and in the future can be configured through API calls. For the time being they are defined and downloaded through Rainforest Cloud by an admin.

```
<UploadSite>
  <Provider>simple_streaming</Provider>
  <Uploader>RFA</Uploader>
  <Description>RFA staging server</Description>
  <HostName>127.0.0.1</HostName>
  <User>0033e4</User>
  <Port>5001</Port>
  <Password>password here</Password>
  <Protocol>http</Protocol>
  <Url>/post_data</Url>
  <Format>RFA</Format>
  <AutoSelect>Y</AutoSelect>
  <Compression>Y</Compression>
  <UploadSize>20</UploadSize>
  <UploadFrequency>600</UploadFrequency>
  <AutoSelect>Y</AutoSelect>
  <Encode>N</Encode>
</UploadSite>
```

Table	Default	Description
Uploader		Simple uploader name (i.e. "Rainforest").
Description		Long description of the upload site.
Hostname		
Protocol	http	http or https
Port	80	Port number (usually 80 or 443, but can be other).
Url		Path part of URL.
X-Header1		Custom header #1 (if required).
X-Header2		Custom header #2 (if required).
X-Header3		Custom header #3 (if required).
Encode	N	Y or N - URL encode POST body?
RequireEmail	N	Upload site required an email address.
RequireUserId	N	
RequirePassword	N	
RequireAuthCode	N	
UploadFrequency	600s	Upload with this frequency. Setting this to "0" will start real-time uploading.
UploadSize	10	Size of upload packets. When this number of packets has been reached, a lump upload will be initiated.
AutoRegister	N	Auto register for this upload site.
AutoSelect	N	Automatically select this uploader. If uploader is de-selected by user, override this. User selection/de-selection will be present in configuration database, "Enabled."
Format		RFA(JSON), XML:PROCESSED and XML:RAW

## 2. Data Types

Here is a list of the various types of data which can be uploaded via the Generic Cloud Uploader.

Item	Units	Description
Demand	kW	Instantaneous demand.
Summation	kWh	Cumulative summation delivered & received.

## 3. Data Format

### 3.1 XML

#### 1. Request

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 timestamp="0000000000s"
version="2.0" macId="0xffffffffffff">
  <XML>
</rainforest>
```

Element	Type, Range, Units	Description
<b>rainforest timestamp</b>	integer string, ms	Milliseconds in Unix time (since Jan 1, 1970)
<b>deviceGuid</b>	hex string, 12 digits	Unique Identifier for the Rainforest device that is posting the upload packet. Ethernet MAC Address is used for EAGLE.
<b>body</b>	array	data samples from devices connected to the Rainforest device

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/xml

<?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. Response

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/1.1 204 No Content
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>
<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.

Each post coming from the gateway is wrapped with the timestamp of the post and the gateway's unique identifier.

## 3.2 JSON

### Request

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

```
.....  
{  
  "timestamp": "1474484326000",  
  "deviceGuid": "d8d5b900355a",  
  "body": [  
  ]  
}
```

## 4. Post Body Formats

### 4.1. DeviceInfo

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

```
<DeviceInfo>
  <DeviceMacId>0xd8d5b9000000b200</DeviceMacId>
  <InstallCode>0xcc15a871ce590351</InstallCode>
  <LinkKey>0xe13e7838b6322f2fca2dcd9c4cf205d8</LinkKey>
  <FWVersion>2.1.6 (9463)</FWVersion>
  <HWVersion>1.3.4</HWVersion>
  <ImageType>0x2101</ImageType>
  <Manufacturer>Rainforest Automation, Inc.</Manufacturer>
  <ModelId>Z114-EAGLE3</ModelId>
  <DateCode>2017040529021003</DateCode>
  <Protocol>Zigbee</Protocol>
</DeviceInfo>
```

Element	Range	Description
<b>DeviceMacId</b>	16 hex digits	MAC Address of <b>EAGLE™</b> ZigBee radio
<b>InstallCode</b>	16 hex digits	Install Code for <b>EAGLE™</b> ZigBee radio
<b>LinkKey</b>	32 hex digits	ZigBee radio Link Key
<b>FWVersion</b>	Text	Firmware Version
<b>HWVersion</b>	Text	Hardware Version
<b>ImageType</b>	4 hex digits	ZigBee code image type
<b>Manufacturer</b>	Text	“Rainforest Automation”
<b>ModelId</b>	Text	“RFA-Z109”
<b>DateCode</b>	YYYYMMDDZZZZZZZZ	Manufacturer’s date code and lot number
<b>Protocol</b>	Text	Source of information

### 4.2. ConnectionStatus

XML:RAW

```
<ConnectionStatus>
  <DeviceMacId>0xd8d5b9000000b200</DeviceMacId>
  <MeterMacId>0x000781000081fd0b</MeterMacId>
  <Status>Rejoining</Status>
  <ExtPanId>0x000781000081fd0b</ExtPanId>
```

```
<Channel>14</Channel>
<ShortAddr>0xd291</ShortAddr>
<LinkStrength>0x00</LinkStrength>
<Protocol>Zigbee</Protocol>
</ConnectionStatus>
```

RFA: We do not upload Connection status in JSON format

### **4.3. TimeCluster**

#### **XML:RAW**

```
<TimeCluster>
  <DeviceMacId>0xd8d5b9000000b1ff</DeviceMacId>
  <UTCTime>0x20f2d7ed</UTCTime>
  <LocalTime>0x20f2d7ed</LocalTime>
  <UTCTimeString>Fri Jul 7, 2017 11:38:21 pm</UTCTimeString>
  <LocalTimeString>Fri Jul 7, 2017 11:38:21 pm</LocalTimeString>
  <Protocol>Zigbee</Protocol>
</TimeCluster>
```

RFA: We do not upload time cluster in JSON format

### **4.4. Instantaneous Demand**

This indicates the power being consumed by a device at a specific point in time.

#### **4.4.1. Body**

##### **RFA**

```
"timestamp": "0",
"subdeviceGuid": "ffffffffffffffff",
"componentId": "00",
"data Type": "InstantaneousDemand",
"data": {
  "demand": 0.0,
  "units": "kW"
}
```

#### **XML:PROCESSED**

```
<InstantaneousDemand>
```

```
<DeviceMacId>0xd8d5b9000000b4a0</DeviceMacId>
<MeterMacId>0x000781000081fd0b</MeterMacId>
<TimeStamp>1498847736</TimeStamp>
<Protocol>Zigbee</Protocol>
<Demand>21.499</Demand>
<Units>kW</Units>
</InstantaneousDemand>
```

## XML:RAW

XML:RAW from the meter:

```
<InstantaneousDemand>
  <DeviceMacId>0xd8d5b9000000b1ff</DeviceMacId>
  <TimeStamp>0x20f03597</TimeStamp>
  <NetworkAddress>0xb2c8</NetworkAddress>
  <IeeeAddress>0x0024460000069489</IeeeAddress>
  <ClusterId>0x0702</ClusterId>
  <SourceEndpoint>0x0b</SourceEndpoint>
  <DestinationEndpoint>0x01</DestinationEndpoint>
  <Protocol>Zigbee</Protocol>
  <Demand>0x00000003</Demand>
  <UnitOfMeasure>0x00</UnitOfMeasure>
  <Multiplier>0x00000001</Multiplier>
  <Divisor>0x000003e8</Divisor>
  <DigitsRight>0x03</DigitsRight>
  <DigitsLeft>0x07</DigitsLeft>
  <SuppressLeadingZero>N</SuppressLeadingZero>
</InstantaneousDemand>
```

Demand from a Subdevice: XML:RAW

```
<ReadAttributesResponse>
  <DeviceMacId>0xd8d5b9000000b6d4</DeviceMacId>
  <TimeStamp>0x20e99d5f</TimeStamp>
  <NetworkAddress>0x8ac8</NetworkAddress>
  <IeeeAddress>0x0024460000068d1c</IeeeAddress>
  <ClusterId>0x0702</ClusterId>
  <SourceEndpoint>0x0c</SourceEndpoint>
  <DestinationEndpoint>0x01</DestinationEndpoint>
  <Attribute1>0x0400</Attribute1>
  <Value1>0x00000000</Value1>
  <Attribute2>0x0300</Attribute2>
  <Value2>0x00</Value2>
  <Attribute3>0x0301</Attribute3>
```

```
<Value3>0x00000001</Value3>
<Attribute4>0x0302</Attribute4>
<Value4>0x000003e8</Value4>
<Attribute5>0x0304</Attribute5>
<Value5>0x3b</Value5>
<Protocol>Zigbee</Protocol>
</ReadAttributesResponse>
```

Demand from a subdevice: XML:PROCESSED

```
<InstantaneousDemand>
  <DeviceMacId>0xd8d5b9000000b6d4</DeviceMacId>
  <TimeStamp>1498866254</TimeStamp>
  <NetworkAddress>0x8ac8</NetworkAddress>
  <IeeeAddress>0x0024460000068d1c</IeeeAddress>
  <ClusterId>0x0702</ClusterId>
  <SourceEndpoint>0x0b</SourceEndpoint>
  <DestinationEndpoint>0x01</DestinationEndpoint>
  <Protocol>Zigbee</Protocol>
  <Demand>0.0</Demand>
  <Units>kW</Units>
</InstantaneousDemand>
```

#### 4.4.2. Example of Single Component Upload

```
{
  "timestamp": "1474484326000",
  "deviceGuid": "d8d5b900355a",
  "body": [{
    "timestamp": "1474484240000",
    "subdeviceGuid": "001bc5007200578f",
    "componentId": "01",
    "dataType": "InstantaneousDemand",
    "data": {
      "demand": 2525.10,
      "units": "kW"
    }
  ]
}
```

Element	Type, Range, Units	Description
<b>timestamp</b>	integer string, ms	Milliseconds in Unix time (since Jan 1, 1970)
<b>subdeviceGuid</b>	hex string, 16 digits	MAC Address of the attached device that is

		measuring the reading
<b>componentId</b>	string	Identifies component within subdevice that the data is coming from
<b>data</b>		list of attribute-value pairs
<b>demand</b>	kW float string, 32bit double	power being used/generated at a moment in time
<b>units</b>	kW	units of the power reading

## 4.5. Current Summation

This indicates the total amount of power consumed by an endpoint since it was first enabled. The number continuously increases unless the device is reset to factory conditions, at which point it will start at zero.

### 4.5.1. Body

RFA

```
"timestamp": "0",
"subdeviceGuid": "ffffffffffffffff",
"componentId": "00",
"dataType": "CurrentSummation ",
"data": {
  "summationDelivered": 0.0,
  "summationReceived": 0.0,
  "units": "kWh"
}
```

XML:RAW:

```
<CurrentSummation>
  <DeviceMacId>0xd8d5b9000000b1ff</DeviceMacId>
  <TimeStamp>0x20f03613</TimeStamp>
  <NetworkAddress>0x7b28</NetworkAddress>
  <IeeeAddress>0x002446000000687a3</IeeeAddress>
  <ClusterId>0x0702</ClusterId>
  <SourceEndpoint>0x0c</SourceEndpoint>
  <DestinationEndpoint>0x01</DestinationEndpoint>
  <Protocol>Zigbee</Protocol>
  <SummationDelivered>0x00000000ec8a31</SummationDelivered>
  <UnitOfMeasure>0x00</UnitOfMeasure>
  <Multiplier>0x00000001</Multiplier>
```

```
<Divisor>0x000003e8</Divisor>
</CurrentSummation>
```

XML:RAW from the meter:

```
<CurrentSummation>
  <DeviceMacId>0xd8d5b9000000af85</DeviceMacId>
  <MeterMacId>0xd8d5b900000021a7</MeterMacId>
  <TimeStamp>0x20acaec0</TimeStamp>
  <SummationDelivered>0x000000000001f81f</SummationDelivered>
  <SummationReceived>0x0000000000000000</SummationReceived>
  <Multiplier>0x00000001</Multiplier>
  <Divisor>0x000003e8</Divisor>
  <UnitOfMeasure>0x00</UnitOfMeasure>
  <DigitsRight>0x03</DigitsRight>
  <DigitsLeft>0x00</DigitsLeft>
  <SuppressLeadingZero>Y</SuppressLeadingZero>
  <Protocol>Zigbee</Protocol>
</CurrentSummation>
```

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

## XML:PROCESSED

```
<CurrentSummation>
  <DeviceMacId>0xd8d5b9000000b4a0</DeviceMacId>
  <MeterMacId>0x000781000081fd0b</MeterMacId>
  <TimeStamp>1498849201</TimeStamp>
  <Protocol>Zigbee</Protocol>
  <SummationDelivered>49808.286</SummationDelivered>
  <SummationReceived>0.0</SummationReceived>
  <Units>kWh</Units>
</CurrentSummation>
```

### 4.5.2. Example of Single Component Upload

```
{
  "timestamp": "1474484326000",
  "deviceGuid": "d8d5b900355a",
  "body": [{
    "timestamp": "1474484343000",
    "subdeviceGuid": "001bc5007200578f",
    "componentId": "01",
    "dataType": "CurrentSummation",
    "data": {
      "summationDelivered": 0.278,
      "summationReceived": 0.69,
      "units": "kWh"
    }
  ]
}
```

Element	Type, Range, Units	Description
<b>timestamp</b>	integer string, ms	Milliseconds in Unix time (since Jan 1, 1970)
<b>subdeviceGuid</b>	hex string, 16 digits	MAC Address of the attached device that is measuring the reading
<b>componentId</b>	string	Identifies component within subdevice that the data is coming from
<b>data</b>		list of attribute-value pairs
<b>summationDelivered</b>	kWh float string, 32bit double	cumulative amount of energy delivered
<b>summationReceived</b>	kWh float string, 32bit double	cumulative amount of energy received

units	kWh	units of the power reading
-------	-----	----------------------------

## 4.6. 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>0xd8d5b9000000af85</DeviceMacId>
  <MeterMacId>0xd8d5b900000021a7</MeterMacId>
  <TimeStamp>0x20acaef</TimeStamp>
  <Id>0x00000000</Id>
  <Text>Welcome to PENGUIN meter simulator.</Text>
  <Priority>Low</Priority>
  <StartTime>0x20acad0d</StartTime>
  <Duration>0xffff</Duration>
  <ConfirmationRequired>N</ConfirmationRequired>
  <Confirmed>N</Confirmed>
  <Queue>Active</Queue>
  <Protocol>Zigbee</Protocol>
</MessageCluster>
```

Element	Range	Description
<b>DeviceMacId</b>	16 hex digits	MAC Address of <b>EAGLE™</b> ZigBee radio
<b>MeterMacId</b>	16 hex digits	MAC Address of Meter
<b>TimeStamp</b>	Up to 8 hex digits	UTC Time (offset in seconds from 00:00:00 01Jan2000) when message was received from meter
<b>Id</b>	Up to 8 hex digits	Message ID from meter
<b>Text</b>	Text	Contents of message, HTML encoded: &gt; replaces the > character &lt; replaces the < character &amp; replaces the & character &quot; replaces the " character
<b>Priority</b>	Low   Medium   High   Critical	Message priority

<b>StartTime</b>		Time when message was retrieved
<b>Duration</b>		
<b>Confirmation Required</b>	Y   N	Y: a user confirmation is required; N: a user confirmation is not required (default)
<b>Confirmed</b>	Y   N	Y: the user confirmation has been sent; N: the user confirmation has not been sent (default)
<b>Queue</b>	Active   Cancel Pending	Active: Indicates message is in active queue Cancel Pending: Indicates message is in cancel pending queue

## 5. Batch Upload

The upload manager can upload data in “batch” format – saving up multiple data readings to upload in a single POST. This could be used in systems where “real time” readings are not important, or if the network connection between the EAGLE and the cloud server is temporarily broken.

### XML:RAW Batch upload:

```
<CurrentSummation>
  <DeviceMacId>0xd8d5b9000000b1ff</DeviceMacId>
  <TimeStamp>0x20f03613</TimeStamp>
  <NetworkAddress>0x7b28</NetworkAddress>
  <IeeeAddress>0x00244600000687a3</IeeeAddress>
  <ClusterId>0x0702</ClusterId>
  <SourceEndpoint>0x0c</SourceEndpoint>
  <DestinationEndpoint>0x01</DestinationEndpoint>
  <Protocol>Zigbee</Protocol>
  <SummationDelivered>0x000000000ec8a31</SummationDelivered>
  <UnitOfMeasure>0x00</UnitOfMeasure>
  <Multiplier>0x00000001</Multiplier>
  <Divisor>0x000003e8</Divisor>
</CurrentSummation>
<InstantaneousDemand>
  <DeviceMacId>0xd8d5b9000000b1ff</DeviceMacId>
  <TimeStamp>0x20f03597</TimeStamp>
  <NetworkAddress>0xb2c8</NetworkAddress>
  <IeeeAddress>0x0024460000069489</IeeeAddress>
  <ClusterId>0x0702</ClusterId>
  <SourceEndpoint>0x0b</SourceEndpoint>
  <DestinationEndpoint>0x01</DestinationEndpoint>
  <Protocol>Zigbee</Protocol>
  <Demand>0x00000003</Demand>
  <UnitOfMeasure>0x00</UnitOfMeasure>
  <Multiplier>0x00000001</Multiplier>
  <Divisor>0x000003e8</Divisor>
  <DigitsRight>0x03</DigitsRight>
  <DigitsLeft>0x07</DigitsLeft>
  <SuppressLeadingZero>N</SuppressLeadingZero>
</InstantaneousDemand>
```

```
{
  "timestamp": "1474484326000",
  "deviceGuid": "d8d5b900355a",
  "body": [{
    "timestamp": "1474484240000",
    "subdeviceGuid": "001bc5007200578f",
    "componentId": "01",
    "dataType": "InstantaneousDemand",
    "data": {
      "demand": 2.0,
      "units": "kW"
    }
  ]
}, {
  "timestamp": "1474484280100",
  "subdeviceGuid": "001bc5007200578f",
  "componentId": "01",
  "dataType": "CurrentSummation",
  "data": {
    "summationDelivered": 0.278,
    "summationReceived": 0.69,
    "units": "kWh"
  }
}]
}
```

## 6. 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_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>`.

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

Element	Range	Description
<b>NetworkInterface</b>	16 hex digits	MAC Address of <b>EAGLE™</b> ZigBee radio
<b>Event</b>	price   summation   demand   message	Type of reading to get from meter
<b>Frequency</b>	4 hex digits	Frequency to poll meter, in seconds
<b>Enabled</b>	Y   N	Y: Set this rate for this type of reading N: Disable polling for this type of reading

### 2. Command: `factory_reset`

```
<Command>
  <Name>factory_reset</Name>
  <NetworkInterface>0xd8d5b9000000b1a0</NetworkInterface>
</Command>
```