

eGauge XML API

(v0.01)

eGauge Systems LLC

November 19, 2010

1 Overview

This document describes how to read raw XML data from an eGauge device using CGI queries. There are two types of queries: instantaneous and stored data queries. The former reads the most recent values of all measured data, whereas the latter reads (portions of) the historical data stored in database built into the eGauge device.

2 Instantaneous Data

Instantaneous data is updated once a second. It is fetched via the URI reference:

```
/cgi-bin/egauge?params
```

The possible values for *params* are described in the next section.

2.1 Query Parameters

Parameter:	Type:	Description:
tot	n/a	Requests that the totals calculated from the register values be included in the output. Each total will appear as a separate meter element. These elements have attribute type set to total so they can be distinguished from regular register meter elements

A sample output for this query is shown in Figure 1.

2.2 Returned XML Data

The instantaneous data query returns a single element enclosed by **measurement** start and end tags. Within the **measurement** element, the following elements may appear:

```
<?xml version="1.0" encoding="UTF-8" ?>
<measurements>
  <timestamp>1284607004</timestamp>
  <cpower src="Grg&Bth (PHEV)" i="11" u="1">-988.9</cpower>
  <cpower src="Solar" i="5" u="8">-1.9</cpower>
  <cpower src="Grid" i="3" u="1">1621.5</cpower>
  <meter title="Grid">
    <energy>1443.5</energy>
    <energyWs>5196771697</energyWs>
    <power>2226.2</power>
  </meter>
  <meter title="Solar">
    <energy>5918.9</energy>
    <energyWs>21308130148</energyWs>
    <power>-3.5</power>
  </meter>
  <meter title="Grg&Bth (PHEV)">
    <energy>4889.2</energy>
    <energyWs>17601054087</energyWs>
    <power>-988.9</power>
  </meter>
  <frequency>59.98</frequency>
  <voltage>119.0</voltage>
  <voltage>118.3</voltage>
  <current>5.495</current>
  <current>14.152</current>
  <current>0.223</current>
  <current>0.136</current>
</measurements>
```

Figure 1: Example of instantaneous data.

Element Name:	Type:	Description:
timestamp	Integer (U32)	The device-local time at which the reported measurements were obtained. This is a UNIX timestamp (seconds since January 1st, 1970 UTC).
meter	Struct	There is one meter element per register configured in the device. Attribute title gives the register-name. Three sub-elements may appear for each meter element: energy , energyWs , and power .
energy	Float	An cumulative energy meter value expressed in units of kilo-Watt-hours (kWh). Subtracting two consecutive readings and dividing by the amount of time elapsed between the samples (expressed in hours) gives average power in kilo-Watts (kW).
energyWs	Integer (S64)	Same as energy , but expressed in units of Watt-seconds (Ws).
power	Float	The average power measured for the most recent one-second interval. Note that this may be positive or negative, depending on the direction of the power-flow.
frequency	Float	Frequency in Hertz as measured on one of the configured voltage taps (L1, L2, or L3). It is presently unspecified which of the configured voltage taps is used to measure frequency.
voltage	Float	RMS voltage in Volts. These elements appear in lowest to highest voltage-channel order.
current	Float	RMS current in Amperes. These elements appear in lowest to highest current-channel order.
cpower	Float	The current value of a power component. Each voltage/current product configured for the device gives rise to one component power. Attribute src identifies which register the component contributes to. Attributes i and u identify the voltage and current channels used to calculate this power. The channel-assignment is device-specific and left unspecified by this document.

3 Stored Data

Stored data is updated once a minute. It is fetched via the URI reference:

```
/cgi-bin/egauge-show?params
```

This query returns energy data as rows of columns. Each row reports data for a specific point in time. The row consists of a fixed number of columns, with one column per configured register. Various query parameters *params* can be specified to select which data to retrieve and what format to return it in.

A sample output for this query using parameters *m&n=3* is shown in Figure 2.

3.1 Returned XML Data

The stored data query returns a single element enclosed by **group** start and end tags. Within the **group** element, the following elements may appear:

Element Name:	Type:	Description:
---------------	-------	--------------

data	Struct	One such element appears for each consecutive sequence of data rows. Attribute columns specifies the number of columns in each row. Attribute time_stamp specifies the UNIX timestamp (in hex) for the first row. Attribute time_delta specifies the number of seconds to be subtracted to get the next row's timestamp. Attribute epoch specifies the UNIX timestamp (in hex) of the time at which recording started. Attribute delta is equal to true if the data rows are delta-encoded (see below). Attribute wait_time specifies how many seconds have to elapse before the timestamp specified by the w parameter is available for reading.
cname	String	Specifies the register-name of a column in order of increasing column. This element may only appear in the first data element. In subsequent data elements, the register names must remain the same as for the first one.
r	Struct	Contains one row of data.
c	Integer (S64)	An individual energy value expressed in units of Watt-seconds (Ws). This value must be divided by 3.6e6 to obtain an energy value in kilo-Watt-hours (kWh). Note that JavaScript cannot natively handle 64-bit values and care must be taken to avoid overflows. These energy values are circular: after reaching the maximum positive value, they wrap around to negative values and vice versa.

3.2 Query Parameters

Parameter:	Type:	Description:
a	n/a	Requests that the totals calculated from the register values be included as the first two columns in each row. Totals are calculated according to the Totaling Rules configured for the device.
b	n/a	Requests the output be returned in the data backup format.
c	n/a	Requests the output be returned in CSV (comma-separated value) format.
m	n/a	Specifies that n and s parameters are specified in units of minutes.
h	n/a	Specifies that n and s parameters are specified in units of hours.
d	n/a	Specifies that n and s parameters are specified in units of days.
C	n/a	Specifies that the returned data be delta-compressed. That is, after the first row of data, each subsequent row's columns are expressed as a difference relative to the previous row's column-values.
n	Integer (U32)	Specifies the maximum number of rows to be returned.
s	Integer (U32)	Specifies the number of rows to skip after outputting a row. For example, $h \& s = 23$ would skip 23 hours worth of data after a row is output, and would be equivalent to d .
f	Integer (U32)	Specifies the timestamp of the first row to be returned.
t	Integer (U32)	Specifies the timestamp of the last row to be returned.
w	Integer (U32)	Specifies the timestamp of the first row to be returned. If the timestamp lies in the future, the query will complete immediately returning an empty data element whose wait_time attribute indicates how many seconds have to elapse before the desired row is available.
T	Integer-list (U32)	Specifies a list of timestamps, ordered by decreasing value (younger to older) for which to return data rows.

z	string	Specifies the time-zone to use when exporting CSV data. The format of this string is described at http://www.opengroup.org/onlinepubs/009695399/basedefs/xbd_chap08.html under environment variable TZ .
---	--------	--

```
<?xml version="1.0" encoding="UTF-8" ?>
<group>
<data columns="3" time_stamp="0x4c9197e4" time_delta="60" epoch="0x47395980">
  <cname>Grid</cname>
  <cname>Solar</cname>
  <cname>Grg&Bth (PHEV)</cname>
  <r><c>5203642184</c><c>21308125431</c><c>17598056700</c></r>
  <r><c>5203503484</c><c>21308125526</c><c>17598116405</c></r>
  <r><c>5203368999</c><c>21308125626</c><c>17598176060</c></r>
</data>
</group>
```

Figure 2: Example of stored data.