Open Hardware Monitor

1 Preface and document conventions

This documented is intended for those who wish to utilise the information exposed to WMI by the Open Hardware Monitor. Its intentions are to provide some basic understanding of some of the structures used, mainly those which are relatively obscure.

2 Type definitions

2.1 HardwareType

```
public enum HardwareType {
  Mainboard,
  SuperIO,
  CPU,
  GpuNvidia,
  GpuAti,
  TBalancer,
  Heatmaster,
  HDD,
}
```

HardwareType is an enumeration intended to distinguish various piece of hardware from each other. One cannot rely on a specific type to indicate the presence of specific sensors, if any at all. This is merely an indication of hardware type in order to be able to potentially use different styles based upon the hardware type. The individual values will not be detailed in this matter, because the names themselves are very clear, and as said, hold no further meaning.

2.2 SensorType

```
public enum SensorType {
  Voltage,
  Clock,
  Temperature,
  Load,
  Fan,
  Flow,
  Control,
  Level
}
```

SensorType is an enumeration to distinguish between various sensors. From a sensor's type, one should derive which display format is to be used, as each type of sensor has a logical unit attached.

Sensor Type	Unit used	Recommended display format
Voltage	Volt	«value» V
Clock	Megahertz	«value» MHz
Temperature	Celsius	«value» °C
Load	Percentage	«value»%
Fan	Revolutions per minute	«value» RPM
Flow	Liters per hour	«value» L/h
Control	Percentage	«value»%
Level	Percentage	«value»%

A few of these types require extra attention, as they are intended to be used under very specific circumstances.

The ***Load***, ***Control***, and ***Level*** types each use percentages as their unit, but have distinctly different purposes.

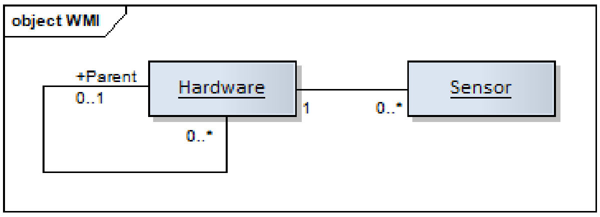
- Load: Represents the load on a piece of hardware, where 100% represents a fully loaded component.
- Control: Typically used in conjunction with fan controllers, used to represent the duty cycle.
- Level: This is a generic % based level measure, used when the above both don't apply.

The ***flow*** type is currently only used in conjunction with the Alphacool Heatmaster and mCubed's T-Balancer bigNG. These two fan controllers have flow sensors for those who have liquid cooled systems and wish to monitor the volumetric flow rate through the system.

3 The WMI provider

3.1 Basics

The Open Hardware Monitor publishes two distinct WMI classes, as detailed below. The namespace they are published to is called "*root/OpenHardwareMonitor*". The cardinality between the 2 classes is defined as follows:



That is: Each sensor instance has exactly one matching parent hardware instance, but a hardware instance does not necessarily have any sensor instances attached to it.

Hardware may have a maximum of one matching parent hardware instance, but a hardware instance does not necessarily have any subhardware instances.

3.2 Hardware

Value type	Name	Description	Example
string	Name	Typically a combination of make and model	Intel Core 2 Duo E8400
string	Identifier	A unique identifier for this piece of hardware	/intelcpu/0
string	HardwareType	The HardwareType as defined in chapter 2.1	CPU
string	Parent	The identifier of its parent Hardware; if any.	"/mainboard" or ""

The identifier includes a zero-based index in order to distinguish between identical pieces of hardware in a single system. This is common in multi-CPU and multi-GPU setups.

The parent identifier will be an empty string if there is no parent (as there won't be for /mainboard for example).

The identifiers are unique per instance, but do not guarantee that the underlying hardware is 100% identical to a previous session. Taking the example of the Intel CPU above, a user could exchange this Intel Core 2 Duo E8400 for an Intel Core 2 Quad Q6600. The identifier will still be /intelcpu/0, but the underlying hardware has changed dramatically. Identifiers intend to identify a specific type of hardware, not the make and model. One has to take this into consideration when working with them, in order to avoid unexpected behaviour.

3.3 Sensor

Value type	Name	Description	Example
string	Name	Typically relative to its parent	CPU Core #2
string	Identifier	A unique identifier for this Sensor	/intelcpu/0/load/1
string	SensorType	The SensorType as defined in chapter 2.2	Load
string	Parent	The identifier of its parent Hardware	/intelcpu/0
float	Value	Last read value	50
float	Min	Lowest read value	0
float	Max	Highest read value	100
int	Index	Index within the array of sensors of the same type	1

The identifier includes a zero-based index in order to distinguish between similar sensors associated with a single piece of hardware. This is common in multi-core CPUs for examples, where each core can have its own ***Load*** or ***Temperature*** sensor.

The identifiers are unique per instance, but do not guarantee that the underlying hardware is 100% identical to a previous session. Taking the example of the Intel CPU above, a user could exchange this Intel Core 2 Duo E8400 for an Intel Core 2 Quad Q6600. The underlying hardware has changed dramatically, but where there were only 2 Load sensors before, there now are 4. Identifiers intend to identify a specific type of hardware, not the make and model. One has to take this into consideration when working with them, in order to avoid unexpected behaviour.

One has to take extra care with the ***Min***, ***Max*** and ***Value*** properties. These are all of the *float* type, and as such will have to be parsed by the client before one can use them for calculations. One also has to take the unit associated with the ***SensorType*** into consideration.