



Redfish Control schema and Sensor enhancement proposal

DMTF Redfish Forum
v0.5 - October 2020

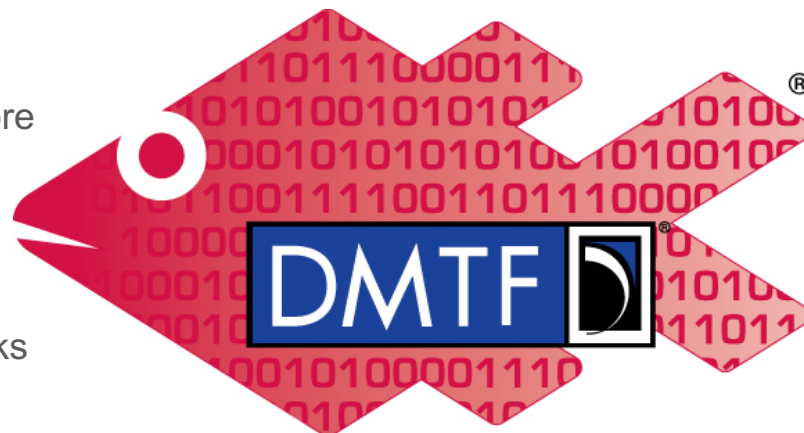
Disclaimer

- The information in this presentation represents a snapshot of work in progress within the DMTF.
- This information is subject to change without notice. The standard specifications remain the normative reference for all information.
- For additional information, see the Distributed Management Task Force (DMTF) website.



Getting involved in Redfish

- Redfish Standards page
 - Schemas, Specs, Mockups, White Papers & more
 - <http://www.dmtf.org/standards/redfish>
- Redfish Developer Portal
 - Redfish Interactive Resource Explorer
 - Educational material, documentation & other links
 - <http://redfish.dmtf.org>
- Redfish User Forum
 - User forum for questions, suggestions and discussion
 - <http://www.redfishforum.com>
- DMTF Feedback Portal
 - Provide feedback or submit proposals for Redfish standards
 - <https://www.dmtf.org/standards/feedback>
- DMTF Redfish Forum
 - Join the DMTF to get involved in future work
 - <http://www.dmtf.org/standards/spmf>



Redfish

Introduction

- General need to represent user or system-owned control points in a system
- These are typically coupled with one or more sensors that provide readings and feedback into a control loop
 - Creating a generic data model for a control, like Redfish already did for sensor, allows software re-use and flexibility
- A “control” includes both the actual setting (the “knob” a client can turn to adjust the setting), and static ‘metadata’ about the control, including:
 - Type of control, units of measure
 - Ranges of supported values (min/max)
 - Minimum increments, accuracy, dead-band
 - Physical location, related equipment
- Like a sensor, there is desire to display a small, primary set of data about the control in context of the device or system it controls, while also having the ability to retrieve the static ‘metadata’ about the control



Goals

- Leverage Sensor model to provide similar designs and usage
- Support controls that directly impact a setting/reading or that provide a desired set point
 - E.g. a “volume” control vs. a “thermostat” – Volume directly impacts a setting, thermostat sets the desired temperature
 - (Volume is a poor example here as it’s unlikely anyone needs metadata about a simple knob like this)
- Provide a “lightweight” implementation option that does not require additional resources for simple controls
 - But apply a consistent model so that implementations can expose more complex instances where desired

Definition for context – “Managed Element”

- Used here as a generic term for the device being managed
 - Examples: ComputerSystem, PowerDistributionUnit, etc.
- A Redfish resource for a managed element will have many properties
 - General information about the element
 - PartNumber, SerialNumber, etc.
 - Status – State, Health, HealthRollup
 - Links to additional (subordinate) resources that describe subsystems
 - May have one or more controls
 - May have one or more associated sensors
- Desire to provide summary of control settings and sensors readings
 - Redfish models attempt to provide this with a single GET
 - Define schemas “excerpt” of the Control and Sensor resources
 - These provide links to Sensor and Control resources to obtain details



CONTROL SCHEMA

Control schema proposal

- New schema heavily leveraged from Sensor
- Describes an individual control point plus associated sensor(s) that measures the effects of that control point
 - More formally known as an “effector”
- *SetPoint* is the primary property for a Control
 - The desired value for a Reading resulting from the Control setting
- *ControlLoop* object to expose PID coefficients for a control loop
 - Additional control methodologies can be added
- Includes a “pass-through” of a single *Reading* from a Sensor
 - Sensor *Reading* might use different measurement units than the *SetPoint*
 - In this case, *SetPointUnits* and *ReadingUnits* are included for clarity
 - Example: Liquid flow valve (percent open), with sensor Reading in liters/second
 - Controls with multiple associated sensors can use separate Sensor excerpts in (managed element) context for more complex relationships

Control Definition (example w/single associated Sensor)

```
{
  "@odata.type": "Control.v1_0_0.Control",
  "@odata.id": "/redfish/v1/Chassis/1U/Controls/Thermostat",
  "Id": "Thermostat",
  "ControlType": "Temperature",
  "ControlUnits": "Cel",
  "SetPoint": 27,
  "DeadBand": 1.5,
  "SetPointUnits": "Cel",
  "OperatingMode": "Automatic",
  "Sensor": {
    "Reading": 27,
    "DataSourceUri": "/redfish/v1/Chassis/1U/Sensors/CabinetTemp"
  },
  "Algorithm": "Average",
  "SetPointRangeMin": 10,
  "SetPointRangeMax": 35,
  "Increment": 0.5,
  "Accuracy": 0.25,
  "ControlLoop": {
    "Proportional": 0.232,
    "Integral": 0.095,
    "Derivative": 0.049
  },
  "Oem": {}
}
```

SetPoint is the primary property of a Control

OperatingMode allows for “Manual” or “Automatic” operation – definition and usage may depend on the type of Control

Reading from a Sensor instance (a Sensor excerpt)

Algorithm could allow model to expose how the control loop operates

ControlLoop would allow model to expose coefficients for a PID-based system, or perhaps allow modification

Control Definition (multiple associated Sensors example)

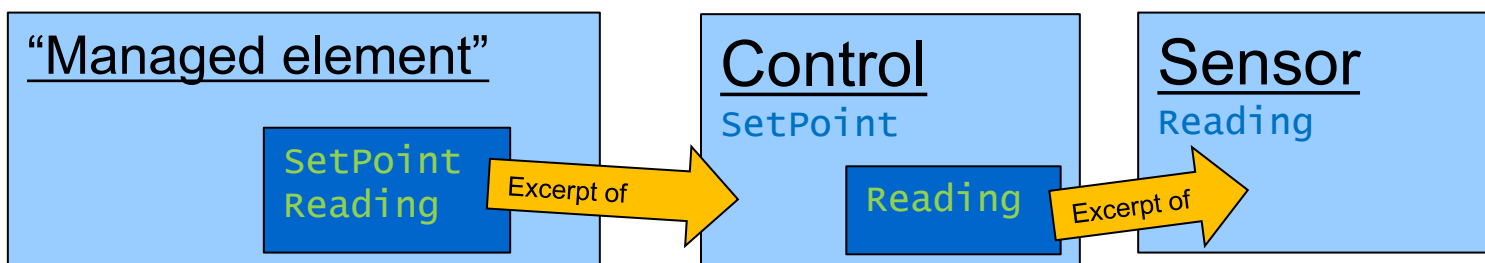
```
{
  "@odata.type": "Control.v1_0_0.Control",
  "@odata.id": "/redfish/v1/Chassis/1U/Controls/Thermostat",
  "Id": "Thermostat",
  "ControlType": "Temperature",
  "SetPointUnits": "Cel",
  "SetPoint": 27,
  "DeadBand": 1.5,
  "ControlDelaySeconds": 180,
  "OperatingMode": "Automatic",
  "SetPointRangeMin": 10,
  "SetPointRangeMax": 35,
  "Increment": 0.5,
  "Accuracy": 0.25,
  "Links": {
    "AssociatedSensors": [
      { "@odata.id": "/redfish/v1/Chassis/1U/Sensors/CabinetTemp" },
      { "@odata.id": "/redfish/v1/Chassis/1U/Sensors/CabinetIntake" },
      { "@odata.id": "/redfish/v1/Chassis/1U/Sensors/CabinetExhaust" }
    ]
  },
  "Oem": {}
}
```

Array of links to multiple Sensor instances instead of a single Sensor excerpt

Control usage of schema excerpts

- An excerpt inserts a copy of certain properties from another resource
 - Allows high-value properties to appear in managed element resources
 - Enables consistent definitions for Sensors and Controls across the data model
 - Also provides a link to the original source of the data
 - Excerpted properties are defined using annotations in Redfish schema
- Control can include a Sensor excerpt
 - Provides *Reading* from a single associated Sensor
 - This property can be further reflected into a managed element resource

Sensor and Control excerpt flow



- Sensor resource
 - *Reading* is defined as an excerpt property
- Control resource
 - Sensor excerpt appears as an object, with *Reading* included
 - *SetPoint* is defined as an excerpt property
 - *DataSourceUri* points to the associated Sensor that provided *Reading*
- "Managed element" resources
 - Control excerpts appears as objects, with *SetPoint* included
 - Also includes *Reading* from the Control's Sensor excerpt as a "pass-through"
 - *Reading* might be rendered as *Reading*<units> to match the Control needs
 - *DataSourceUri* points to the associated Control

Control excerpt examples

- An excerpt appears in a managed element resource as an object containing the excerpted properties and a link to source of that data

```
"ThermostatCelsius": {  
  "Reading": 27,  
  "SetPoint": 27,  
  "DataSourceUri": "/redfish/v1/Chassis/1U/Controls/Thermostat"  
},
```

Example #1: *SetPoint* has the same units as *Reading* from a single Sensor instance

```
"CoolingWaterLoopControl": {  
  "Reading": 7.93,  
  "ReadingUnits": "GPM",  
  "SetPoint": 70,  
  "SetPointUnits": "%",  
  "DataSourceUri": "/redfish/v1/Chassis/1U/Controls/Valve1"  
},
```

Example #2: *SetPoint* and *Reading* use different units. Excerpt includes *SetPointUnits* and the *ReadingUnits* property from Sensor to clarify the difference in units.

DataSourceUri points to the Control resource. The Control resource will contain a Sensor excerpt with *DataSourceUri* further pointing to the Sensor resource.

Complex Control example

- Multiple excerpts appear in this managed element resource as objects containing the excerpted properties and links to each data source
- This provides a compact summary of the element in a single resource

```
{
  "WidgetControl": {
    "ComplexControlPercent": {
      "SetPoint": 75,
      "DataSourceUri": "/redfish/v1/Chassis/1U/Controls/PrimaryValve"
    },
    "LiquidFlowLiters": {
      "Reading": 7.93,
      "DataSourceUri": "/redfish/v1/Chassis/1U/Sensors/PrimaryValveFlow"
    },
    "InputPressurePsi": {
      "Reading": 35.81,
      "DataSourceUri": "/redfish/v1/Chassis/1U/Sensors/IntakePressure"
    },
    "OutputPressurePsi": {
      "Reading": 11.24,
      "DataSourceUri": "/redfish/v1/Chassis/1U/Sensors/OutflowPressure"
    }
  }
}
```

This Control has multiple Sensor associations

Control excerpt contains the *SetPoint*

Three Sensor excerpts show the resulting values as *Reading*

Excerpt object names include Reading or SetPoint units

Fan resource concept

- “Fan” is an example of a managed element resource
- Contains a Sensor excerpt, and may expose a Control as well

```
{  
  "SpeedControlPWM": {  
    "SetPoint": 125,  
    "DataSourceUri": "/redfish/v1/Chassis/1U/Controls/FanBay1"  
  },  
  "SpeedPercent": {  
    "Reading": 55,  
    "SpeedRPM": 2300,  
    "DataSourceUri": "/redfish/v1/Chassis/1U/Sensors/FanBay1"  
  }  
}
```

Control excerpt to show SetPoint, if supported

Sensor excerpt to show Reading and
“extra fan excerpt property” SpeedRPM

Fan instances that expose the control include the *SpeedControlPWM* object (excerpt of Control), while fans without the control exposed include only the *SpeedPercent* object (excerpt of Sensor)



SENSOR ENHANCEMENTS



User-defined Thresholds

- Add User-defined Threshold support in Sensor
 - Clearly define service vs user-defined thresholds
 - Implementation may support user-defined thresholds for each sensor
 - Some existing *Thresholds* usage may move to *UserThresholds*
 - If user can define reaction behavior (but perhaps not change value)?
- New sub-objects added to *Thresholds* object with “user” naming
 - *UpperCautionUser*, *UpperCriticalUser*, *LowerCautionUser*, *LowerCriticalUser*

Additional Sensor Reading types

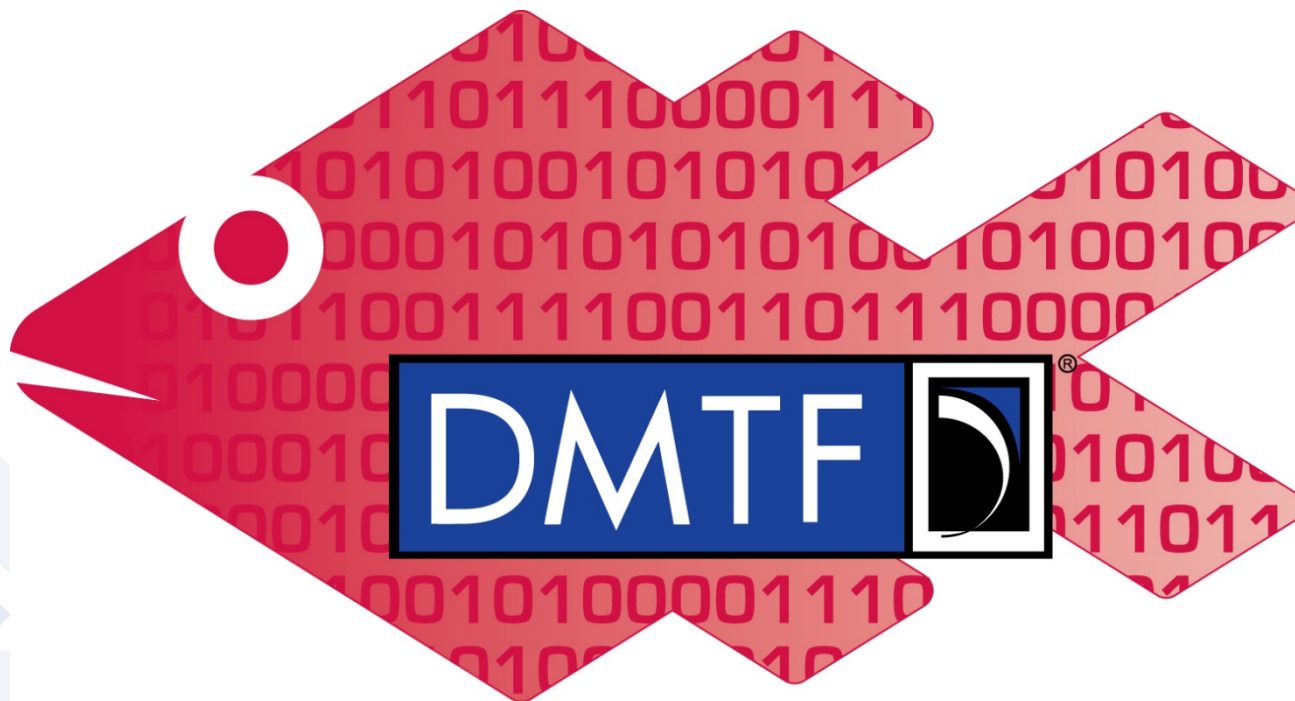
Reading (Sensor) Type	Unit of Measure
Weight	Kilogram
Force	newton-meter
Luminosity	lux
Sound	dB (A-weighted or C-weighted)
LinearPosition	meter
AngularPosition	radian
Velocity	meters per second

Control links

- Add Controls links to show relationships
 - “What control(s) do I use to affect changes to this reading?”
 - Provide array of Links to locate Controls that affect the sensor

```
“Links”: {  
  “Controls”: [{  
    { “@odata.id”: “/redfish/v1/Chassis/1U/Controls/PrimaryValve“ },  
    { “@odata.id”: “/redfish/v1/Chassis/1U/Controls/SecondaryValve“ }  
  ]  
},
```

Q&A & Discussion



Redfish