Document Number: DSP2049

Date: 2017-05-30

Version: 0.2.2b

OCP Hardware Management with Redfish

Information for Work-in-Progress version:

IMPORTANT: This document is not a standard. It does not necessarily reflect the views of the DMTF or its members. Because this document is a Work in Progress, this document may still change, perhaps profoundly and without notice. This document is available for public review and comment until superseded.

Provide any comments through the DMTF Feedback Portal: http://www.dmtf.org/standards/feedback

Supersedes: None

Document Class: Informative

Document Status: Work in Progress

Document Language: en-US

Contents

Figures

Tables

OCP Hardware Management with Redfish

1 Scope

This document defines the Redfish model to remotely manage platforms and devices in Open Compute Project.

2 Overview

Scalability in today's data center is increasingly achieved with horizontal, scale-out solutions, which often include large quantities of simple servers. The usage model of scale-out hardware is drastically different than that of traditional enterprise platforms, and requires a new approach to management.

Designed to meet the expectations of end users for simple and secure management of modern scalable platform hardware, DMTF's Redfish® is an open industry standard specification and schema that specifies a RESTful interface and utilizes JSON and OData to help customers integrate solutions within their existing tool chains. An aggressive development schedule is quickly advancing Redfish toward its goal of addressing all the components in the data center with a consistent API.

Redfish is composed of an interface specification and resource models, for different management domains. The resource models are specified as schema, in OData CSDL (Common Schema Descriptor Language) and json-schema format. Mockups of the resources have been found to be more easily understood and are present in the document.

3 Redfish Interface Features

3.1 Introduction

This document specifies the Redfish model elements that support OCP Hardware Management. This document describes the Redfish equivalent mechanism to support OCP Hardware Management.

The OCP Hardware Management working group has posted multiple document drafts specifying how the OCP Hardware Manage can be supported with IPMI (Intelligent Platform Management Interface).

The "Open Hardware Management v1.01" draft describes the use cases and the IPMI-based commands required to support remote machine management. Other IPMI-related documents are in various stages of draft.

This document starts with the OCP Hardware Management v1.01 document and specifies the equivalent Redfish mechanism. For the other IPMI-related drafts, this document attempts to specify the equivalent Redfish mechanism, as much as possible.

3.2 Reference Documents

3.2.1 Redfish

The DMTF has various locations for learning and communicating about Redfish

- Dmtf.org/redfish: Releases of updates to the Redfish Specification and schema
- Redfish.dmtf.org: Redfish Developer Hub (training, open-source tools, etc.)
- Redfishforum.com: Public community forum (Redfish, Swordfish, Client SW, Service Implementations, etc.)

Redfish Interface Specification v1.1 (DSP0266)
http://www.dmtf.org/sites/default/files/standards/documents/DSP0266_1.1.0.pdf

An interactive explorer of the OCP Redfish Profile discussed in the document is available at http://redfish.dmtf.org/redfish/v1/mockup/770

3.2.2 OCP Hardware Management

The OCP Hardware Management specifications can be found on their Wiki.

http://www.opencompute.org/wiki/Hardware_Management/SpecsAndDesigns

- DRAFT - Open Hardware Management v1.01 (Feb 2014)

This specification specifies baseline set of commands, which references the DCMI specification. Similarly, the DCMI specification reference commands in the IPMI specification.

- DRAFT - Requirements for Firmware Update, v0.2b (Aug 2014)

This document specifies the firmware update requirements for OCP compliant platforms and devices.

- DRAFT - OCP Hardware Management Specifications for IPMI

This section list four specifications in early draft, which specify the IPMI implementations on any device using IPMI.

- Hardware Management - ICAP BASE, Version 0.02 (June 2014)
- Hardware Management - ICAP DRAM, Version 0.03 (June 2014)
- Hardware Management - ICAP Optical, Version 0.02 (June 2014)
- Hardware Management - SPEC ID, Version 0.04 (June 2014)

- Cloud Server Multi Node System Specification v0.7.5 (Aug 2015)

This document describes the requirement of a Cloud Server (server, enclosure, rack).

4 OCP Redfish Profile

4.1 Overview

The profile supports basic management features, as documented in OCP Hardware Management Specification for IPMI v1.01

The profile supports a single monolithic server platform.

- One ComputerSystem
- One Chassis
- One Manager

The profile supports the following management features.

- Power-on/off/reset
- Boot to PXE, HDD, BIOS setup (boot override)
- 4 temp sensors per DCMI (CPU1, CPU2, Board, Inlet)
- Simple Power Reading, and DCMI Power Limiting
- Fan Monitoring w/ redundancy
- Set asset tag and Indicator LED

- Basic inventory (serial#, model, SKU, Vendor, BIOS ver…)
- User Management
- BMC management: get/set IP, version, enable/disable protocol

4.2 Interface

The management interface shall conform to the Redfish Specification v1.1 (DSP0266).

4.3 Resources

Table 1 specifies the Redfish resources that could be present in the interface. A resource is singleton resource, unless otherwise specified.

The management interface could support the following resources.

Table 1 - Redfish Base Model

Resource	URI
Service Root	/
Systems collection	/Systems
Computer System	/Systems/{id}
Log Services	/Systems/{id}/LogServices
SEL (System Event Log)	/Systems/{id}/LogServices/SEL
Entry in SEL	/Systems/{id}/LogServices/SEL/Entries/{id}
Chassis collection	/Chassis
Chassis	/Chassis/{id}
Power	/Chassis/{id}/Power
Thermal	/Chassis/{id}/Thermal
Managers collection	/Managers
BMC (baseboard mgmt controller)	/Managers/BMC
BMC's Ethernet interface	/Managers/BMC/EthernetInterfaces/eth0
BMC's network protocol	/Managers/BMC/NetworkProtocol
Account Service	/AccountService
Accounts collection	/AccountService/Accounts
Roles collection	/AccountService/Roles
Session Service	/SessionService
schema	/$metadata
odata	/odata

4.4 Collection resources

In Redfish, the collection resources have the same JSON format. There is a Members property which contains a link to the members of the collection. Since this profile is for a single monolithic server platform, only one member should exist.

Figure 1 is an example of the JSON response to a GET request for a Chassis collection resource.

Figure 1 – Chassis collection resource

```
{
    "@odata.type": "#ChassisCollection.ChassisCollection",
    "Name": "Chassis Collection",
    "Members@odata.count": 1,
    "Members": [
        {
            "@odata.id": "/redfish/v1/Chassis/1"
        }
    ],
    "@odata.context": "/redfish/v1/$metadata#ChassisCollection.ChassisCollection",
    "@odata.id": "/redfish/v1/Chassis",
    "@Redfish.Copyright": ". . ."
}
```

4.5 Computer System resource

Figure 2 is an example of the JSON response to a GET request for a ComputerSystem resource.

This profile could contain each of the properties shown in Figure 2.

The System resource supports a 'Reset' action. The reset is performed by sending a POST request to the path specified in the "target" property. The POST request could include a value for the ResetType property. The reset type values supported by the implement is contained in the ResetType@Redfish.AllowableValues property.

Figure 2 - ComputerSystem resource

```
{
    "@odata.type": "#ComputerSystem.v1_1_0.ComputerSystem",
    "Id": "1",
    "Name": "Catfish System",
    "SystemType": "Physical",
    "AssetTag": "CATFISHASSETTAG",
    "Manufacturer": "CatfishManufacturer",
    "Model": "YellowCat1000",
    "SerialNumber": "2M220100SL",
    "SKU": "",
    "PartNumber": "",
    "Description": "Catfish Implementation Recipe of simple scale-out monolithic
server",
    "UUID": "00000000-0000-0000-0000-000000000000",
    "HostName": "catfishHostname",
    "PowerState": "On",
    "BiosVersion": "X00.1.2.3.4(build-23)",
    "Status": {
        "State": "Enabled",
        "Health": "OK"
    },
    "IndicatorLED": "Off",
    "Boot": {
        "BootSourceOverrideEnabled": "Once",
        "BootSourceOverrideMode": "UEFI",
        "UefiTargetBootSourceOverride": "uefiDevicePath",
        "BootSourceOverrideTarget": "Pxe",
        "BootSourceOverrideTarget@Redfish.AllowableValues": [
            "None",
            "Pxe",
            "Usb",
            "Hdd",
            "BiosSetup",
            "UefiTarget",
            "UefiHttp"
        ]
    },
    "LogServices": {
        "@odata.id": "/redfish/v1/Systems/1/LogServices"
    },
    "Links": {
        "Chassis":   [{ "@odata.id": "/redfish/v1/Chassis/1" }],
        "ManagedBy": [{ "@odata.id": "/redfish/v1/Managers/bmc" }]
    },
    "Actions": {
        "#ComputerSystem.Reset": {
            "target": "/redfish/v1/Systems/1/Actions/ComputerSystem.Reset",
            "ResetType@Redfish.AllowableValues": [
                "On",
                "ForceOff",
                "GracefulShutdown",
                "ForceRestart",
                "Nmi",
                "GracefulRestart",
                "ForceOn"
            ]
        }
    },
    "@odata.context": "/redfish/v1/$metadata#ComputerSystem.ComputerSystem",
    "@odata.id": "/redfish/v1/Systems/1",
    "@Redfish.Copyright": "..."
}
```

4.6 SEL (System Event Log) resource

Figure 3 is an example of the JSON response to a GET request for the SEL singleton resource.

This profile could contain each of the properties shown in Figure 3.

Figure 3 - SEL resource

```
{
    "@odata.type": "#LogService.v1_0_2.LogService",
    "Id": "SEL",
    "Name": "System Log Service",
    "MaxNumberOfRecords": 1000,
    "OverWritePolicy": "WrapsWhenFull",
    "DateTime": "2015-03-13T04:14:33+06:00",
    "DateTimeLocalOffset": "+06:00",
    "ServiceEnabled": true,
    "Status": {
        "State": "Enabled",
        "Health": "OK"
    },
    "Actions": {
        "#LogService.ClearLog": {
            "target": "/redfish/v1/Systems/1/LogServices/SEL/Actions/LogService.Reset"
        }
    },
    "Entries": { "@odata.id": "/redfish/v1/Systems/1/LogServices/SEL/Entries" },

    "@odata.context": "/redfish/v1/$metadata#LogService.LogService",
    "@odata.id": "/redfish/v1/Systems/1/LogServices/SEL",
    "@Redfish.Copyright": "..."
```

4.7 Log Entry resource

Figure 4 is an example of the JSON response to a GET request for the SEL event singleton resource.

This profile could contain each of the properties shown in Figure 4.

Figure 4 – Log Entry resource

```
{
    "@odata.type": "#LogEntry.v1_0_2.LogEntry",
    "Id": "1",
    "Name": "Log Entry 1",
    "EntryType": "SEL",
    "OemRecordFormat": "CompanyX",
    "Severity": "Critical",
    "Created": "2012-03-07T14:44",
    "EntryCode": "Assert",
    "SensorType": "Temperature",
    "SensorNumber": 1,
    "Message": "Message for Event, Description for SEL, OEM depends",
    "MessageId": "Event.1.0.TempAssert",
    "MessageArgs": [ "ArrayOfMessageArgs" ],
    "Links": {
        "OriginOfCondition": { "@odata.id": "/redfish/v1/Chassis/1/Thermal" }
    },

    "@odata.context": "/redfish/v1/$metadata#LogEntry.LogEntry",
    "@odata.id": "/redfish/v1/Systems/1/LogServices/SEL/Entries/1",
    "@Redfish.Copyright":"..."
}
```

4.8 Chassis resource

Figure 5 is an example of the JSON response to a GET request on a Chassis singleton resource.

This profile could contain each of the properties shown in Figure 5.

Figure 5 - Chassis resource

```
{
    "@odata.type": "#Chassis.v1_2_0.Chassis",
    "Id": "1",
    "Name": "Catfish System Chassis",
    "ChassisType": "RackMount",
    "Manufacturer": "CatfishManufacturer",
    "Model": "YellowCat1000",
    "SerialNumber": "2M220100SL",
    "SKU": "",
    "PartNumber": "",
    "AssetTag": "CATFISHASSETTAG",
    "IndicatorLED": "Lit",
    "PowerState": "On",
    "Status": {
        "State": "Enabled",
        "Health": "OK"
    },
    "Thermal": { "@odata.id": "/redfish/v1/Chassis/1/Thermal" },
    "Power":   { "@odata.id": "/redfish/v1/Chassis/1/Power" },
    "Links": {
        "ComputerSystems":   [{ "@odata.id": "/redfish/v1/Systems/1" }],
        "ManagedBy":         [{ "@odata.id": "/redfish/v1/Managers/bmc" }],
        "ManagersInChassis": [{ "@odata.id": "/redfish/v1/Managers/bmc" }]
    },
    "@odata.context": "/redfish/v1/$metadata#Chassis.Chassis",
    "@odata.id": "/redfish/v1/Chassis/1",
    "@Redfish.Copyright": "..."
}
```

4.9 Power resource

Figure 6 is an example of the JSON response to a GET request on a Power singleton resource.

This profile could contain each of the properties shown in Figure 6.

Figure 6 - Power resource

```
{
    "@odata.type": "#Power.v1_1_0.Power",
    "Id": "Power",
    "Name": "Power",
    "PowerControl": [
        {
            "@odata.id": "/redfish/v1/Chassis/1/Power#/PowerControl/0",
            "MemberId": "0",
            "Name": "System Power Control",
            "PowerConsumedWatts": 224,
            "PowerCapacityWatts": 600,
            "PowerLimit": {
                "LimitInWatts": 450,
                "LimitException": "LogEventOnly",
                "CorrectionInMs": 1000
            },
            "Status": {
                "State": "Enabled",
                "Health": "OK"
            }
        }
    ],
    "@odata.context": "/redfish/v1/$metadata#Power.Power",
    "@odata.id": "/redfish/v1/Chassis/1/Power",
    "@Redfish.Copyright": "..."
}
```

4.10 Thermal resource

Figure 7 is an example of the JSON response to a GET request on a Thermal singleton resource.

The Temperatures property could contain an entry for:

- Inlet Temperature
- Board Temperature
- CPU Temperature for each of the processors

The Fans property could contain an entry for each fan present on the platform. If the fans are redundant, the Redundancy property could be present.

The resource could contain each of the remaining properties shown in Figure 7.

Figure 7 - Thermal resource

```
{
    "@odata.type": "#Thermal.v1_1_0.Thermal",
    "Id": "Thermal",
    "Name": "Thermal",
    "Temperatures": [
        {
            "@odata.id": "/redfish/v1/Chassis/1/Thermal#/Temperatures/0",
            "MemberId": "0",
            "Name": "Inlet Temp",
            "SensorNumber": 42,
            "Status": {
                "State": "Enabled",
                "Health": "OK"
            },
            "ReadingCelsius": 25,
            "UpperThresholdNonCritical": 35,
            "UpperThresholdCritical": 40,
            "UpperThresholdFatal": 50,
            "MinReadingRange": 0,
            "MaxReadingRange": 200,
            "PhysicalContext": "Intake"
        },
        {
            "@odata.id": "/redfish/v1/Chassis/1/Thermal#/Temperatures/1",
            "MemberId": "1",
            "Name": "Board Temp",
            "SensorNumber": 43,
            "Status": {
                "State": "Enabled",
                "Health": "OK"
            },
            "ReadingCelsius": 35,
            "UpperThresholdNonCritical": 30,
            "UpperThresholdCritical": 40,
            "UpperThresholdFatal": 50,
            "MinReadingRange": 0,
            "MaxReadingRange": 200,
            "PhysicalContext": "SystemBoard"
        },
        {
            "@odata.id": "/redfish/v1/Chassis/1/Thermal#/Temperatures/2",
            "MemberId": "2",
            "Name": "CPU1 Temp",
            "SensorNumber": 44,
            "Status": {
                "State": "Enabled",
                "Health": "OK"
            },
            "ReadingCelsius": 45,
            "UpperThresholdNonCritical": 60,
            "UpperThresholdCritical": 82,
            "MinReadingRange": 0,
            "MaxReadingRange": 200,
            "PhysicalContext": "CPU"
        },
        {
            "@odata.id": "/redfish/v1/Chassis/1/Thermal#/Temperatures/3",
            "MemberId": "3",
            "Name": "CPU2 Temp",
            "SensorNumber": 45,
            "Status": {
                "State": "Enabled",
                "Health": "OK"
            },
            "ReadingCelsius": 46,
            "UpperThresholdNonCritical": 60,
            "UpperThresholdCritical": 82,
            "MinReadingRange": 0,
            "MaxReadingRange": 200,
            "PhysicalContext": "CPU"
        }
    ],
    "Fans": [
        {
            "@odata.id": "/redfish/v1/Chassis/1/Thermal#/Fans/0",
            "MemberId": "0",
            "Name": "BaseBoard System Fan 1",
            "PhysicalContext": "Backplane",
            "Status": {
                "State": "Enabled",
                "Health": "OK"
            },
            "Reading": 2100,
            "ReadingUnits": "RPM",
            "UpperThresholdNonCritical": 42,
            "UpperThresholdCritical": 4200,
            "UpperThresholdFatal": 42,
            "LowerThresholdNonCritical": 42,
            "LowerThresholdCritical": 5,
            "LowerThresholdFatal": 42,
            "MinReadingRange": 0,
            "MaxReadingRange": 5000,
            "Redundancy": [
                {
                    "@odata.id": "/redfish/v1/Chassis/1/Thermal#/Redundancy/0"
                }
            ]
        },
        {
            "@odata.id": "/redfish/v1/Chassis/1/Thermal#/Fans/1",
            "MemberId": "1",
            "Name": "BaseBoard System Fan 2",
            "PhysicalContext": "Backplane",
            "Status": {
                "State": "Enabled",
                "Health": "OK"
            },
            "Reading": 2100,
            "ReadingUnits": "RPM",
            "UpperThresholdNonCritical": 42,
            "UpperThresholdCritical": 4200,
            "UpperThresholdFatal": 42,
            "LowerThresholdNonCritical": 42,
            "LowerThresholdCritical": 5,
            "LowerThresholdFatal": 42,
            "MinReadingRange": 0,
            "MaxReadingRange": 5000,
            "Redundancy": [
                {
                    "@odata.id": "/redfish/v1/Chassis/1/Thermal#/Redundancy/0"
                }
            ]
        }
    ],
    "Redundancy": [
        {
            "@odata.id": "/redfish/v1/Chassis/1/Thermal#/Redundancy/0",
            "MemberId": "0",
            "Name": "BaseBoard System Fans",
            "RedundancySet": [
                {
                    "@odata.id": "/redfish/v1/Chassis/1/Thermal#/Fans/0"
                },
                {
                    "@odata.id": "/redfish/v1/Chassis/1/Thermal#/Fans/1"
                }
            ],
            "Mode": "N+m",
            "Status": {
                "State": "Enabled",
                "Health": "OK"
            },
            "MinNumNeeded": 1,
            "MaxNumSupported": 2
        }
    ],
    "@odata.context": "/redfish/v1/$metadata#Thermal.Thermal",
    "@odata.id": "/redfish/v1/Chassis/1/Thermal",
    "@Redfish.Copyright": "..."
}
```

4.11 BMC resource

Figure 8 is an example of the JSON response to a GET request on a BMC resource.

The resource could contain each of the properties shown in Figure 8.

The resource could support the Reset action.

Figure 8 - BMC resource

```
{
    "@odata.type": "#Manager.v1_1_0.Manager",
    "Id": "bmc",
    "Name": "Manager",
    "ManagerType": "BMC",
    "Description": "BMC",
    "ServiceEntryPointUUID": "92384634-2938-2342-8820-489239905423",
    "UUID": "00000000-0000-0000-0000-000000000000",
    "Model": "CatfishBMC",
    "DateTime": "2015-03-13T04:14:33+06:00",
    "DateTimeLocalOffset": "+06:00",
    "Status": {
        "State": "Enabled",
        "Health": "OK"
    },
    "FirmwareVersion": "1.00",
    "NetworkProtocol":    { "@odata.id": "/redfish/v1/Managers/bmc/NetworkProtocol" },
    "EthernetInterfaces": { "@odata.id": "/redfish/v1/Managers/bmc/EthernetInterfaces" },
    "Links": {
        "ManagerForServers": [{ "@odata.id": "/redfish/v1/Systems/1" }],
        "ManagerForChassis": [{ "@odata.id": "/redfish/v1/Chassis/1" }],
        "ManagerInChassis":   { "@odata.id": "/redfish/v1/Chassis/1" }
    },
    "Actions": {
        "#Manager.Reset": {
            "target": "/redfish/v1/Managers/bmc/Actions/Manager.Reset",
            "ResetType@Redfish.AllowableValues": [
                "ForceRestart",
                "GracefulRestart"
            ]
        }
    },

    "@odata.context": "/redfish/v1/$metadata#Manager.Manager",
    "@odata.id": "/redfish/v1/Managers/bmc",
    "@Redfish.Copyright": "..."
}
```

4.12 BMC Ethernet Interface resource

Figure 9 is an example of the JSON response to a GET request on a BMC Ethernet interface resource.

The resource could contain each of the properties shown in Figure 9.

Figure 9 - BMC Ethernet interface resource

```
{
    "@odata.type": "#EthernetInterface.v1_0_2.EthernetInterface",
    "Id": "eth0",
    "Name": "Manager Ethernet Interface",
    "Description": "Management Network Interface",
    "Status": {
        "State": "Enabled",
        "Health": "OK"
    },
    "InterfaceEnabled": true,
    "PermanentMACAddress": "AA:BB:CC:DD:EE:FF",
    "MACAddress": "AA:BB:CC:DD:EE:FF",
    "SpeedMbps": 100,
    "AutoNeg": true,
    "FullDuplex": true,
    "MTUSize": 1500,
    "HostName": "MyHostName",
    "FQDN": "MyHostName.MyDomainName.com",
    "MaxIPv6StaticAddresses": 1,
    "VLAN": {
        "VLANEnable": true,
        "VLANId": 101
    },
    "IPv4Addresses": [
        {
            "Address": "192.168.0.10",
            "SubnetMask": "255.255.252.0",
            "AddressOrigin": "DHCP",
            "Gateway": "192.168.0.1"
        }
    ],
    "IPv6AddressPolicyTable": [
        {
            "Prefix": "::1/128",
            "Precedence": 50,
            "Label": 0
        }
    ],
    "IPv6StaticAddresses": [
        {
            "Address": "fe80::1ec1:deff:fe6f:1e24",
            "PrefixLength": 16
        }
    ],
    "IPv6DefaultGateway": "fe80::1ec1:deff:fe6f:1e24",
    "IPv6Addresses": [
        {
            "Address": "fe80::1ec1:deff:fe6f:1e24",
            "PrefixLength": 64,
            "AddressOrigin": "SLAAC",
            "AddressState": "Preferred"
        }
    ],

    "@odata.context": "/redfish/v1/$metadata#EthernetInterface.EthernetInterface",
    "@odata.id": "/redfish/v1/Managers/bmc/EthernetInterfaces/eth0",
    "@Redfish.Copyright":"..."
```

4.13 BMC Network Protocol resource

Figure 10 is an example of the JSON response to a GET request on a BMC network protocol resource.

The resource could contain each of the properties shown in Figure 10.

Figure 10 - BMC network protocol resource

```
{
    "@odata.type": "#ManagerNetworkProtocol.v1_0_2.ManagerNetworkProtocol",
    "Id": "NetworkProtocol",
    "Name": "Manager Network Protocol",
    "Description": "Manager Network Service Status",
    "Status": {
        "State": "Enabled",
        "Health": "OK"
    },
    "HostName": "myBmcHostname",
    "FQDN": "mymanager.mydomain.com",
    "HTTP":   { "ProtocolEnabled": true, "Port": 80 },
    "HTTPS":  { "ProtocolEnabled": true, "Port": 443 },
    "IPMI":   { "ProtocolEnabled": true, "Port": 623 },
    "SSH":    { "ProtocolEnabled": true, "Port": 22 },
    "SNMP":   { "ProtocolEnabled": true, "Port": 161 },
    "Telnet": { "ProtocolEnabled": true, "Port": 23 },
    "SSDP": {
        "ProtocolEnabled": true,
        "Port": 1900,
        "NotifyMulticastIntervalSeconds": 600,
        "NotifyTTL": 5,
        "NotifyIPv6Scope": "Site"
    },

    "@odata.context": "/redfish/v1/$metadata#ManagerNetworkProtocol.ManagerNetworkProtocol",
    "@odata.id": "/redfish/v1/Managers/bmc/NetworkProtocol",
    "@Redfish.Copyright":"..."
}
```

4.14 Account resource

Figure 11 is an example of the JSON response to a GET request on an Account resource.

The resource could contain each of the properties shown in Figure 11.

Figure 11 – Account resource

```
{
    "@odata.type": "#ManagerAccount.v1_0_2.ManagerAccount",
    "Id": "root",
    "Name": "UserAccount",
    "Description": "User Account",
    "Enabled": true,
    "Password": null,
    "UserName": "root",
    "RoleId": "Administrator",
    "Locked": false,
    "Links": {
        "Role": { "@odata.id": "/redfish/v1/AccountService/Roles/Admin" }
    },
    "@odata.context": "/redfish/v1/$metadata#ManagerAccount.ManagerAccount",
    "@odata.id": "/redfish/v1/AccountService/Accounts/root",
    "@Redfish.Copyright":"..."
}
```

4.15 Role resource

Figure 12 is an example of the JSON response to a GET request on a Role resource.

The resource could contain each of the properties shown in Figure 12.

Figure 12 – Role resource

```
{
    "@odata.type": "#Role.v1_0_2.Role",
    "Id": "Administrator",
    "Name": "User Role",
    "Description": "Admin User Role",
    "IsPredefined": true,
    "AssignedPrivileges": [
        "Login",
        "ConfigureManager",
        "ConfigureUsers",
        "ConfigureSelf",
        "ConfigureComponents"
    ],
    "@odata.context": "/redfish/v1/$metadata#Role.Role",
    "@odata.id": "/redfish/v1/AccountService/Roles/Admin",
    "@Redfish.Copyright":"..."
}
```

4.16 Session resource

Figure 13 is an example of the JSON response to a GET request on a Session resource.

The resource could contain each of the properties shown in Figure 13.

Figure 13 – Session resource

```
{
    "@odata.type": "#Session.v1_0_2.Session",
    "Id": "1234567890ABCDEF",
    "Name": "User Session",
    "Description": "Manager User Session",
    "UserName": "root",

    "@odata.context": "/redfish/v1/$metadata#Session.Session",
    "@odata.id": "/redfish/v1/SessionService/Sessions/1234567890ABCDEF",
    "@Redfish.Copyright":"..."
}
```

5 Open Hardware Management v1.01

5.1 Mapping

The mapping below is organized according to the IPMI-based documents. The IPMI column contains the referenced IPMI-based command.

The Redfish equivalent column contains the mechanism that would be used to perform the same task.

In some cases, the task can be described simply. In those cases, the column contains the HTTP request, the generalized path and the action that the client would perform on the JSON included in the HTTP response.

In this description text, whether the resource and/or property currently exists in the Redfish model is indicated by the color of the text:

- Black text – resource or property exists in the Redfish model
- Blue text – resource or property does not exists in the Redfish model. The text is a proposal.

Table 2 contains the Redfish model elements to support hardware management. Redfish supports a collection of Managers and each managers is a singleton resource (./Managers/{id}). For a platform with only one manager, a BMC (baseboard management controller), then {id} = BMC.

Table 2- Redfish Base Model

IPMI Command	Redfish Equivalent	In OCP mockup
Get DCMI Capabilities Info	GET ./Profiles…. (is this needed)???	
Set & Get DCMI Configuration Parameters	TBD	N
Get Management Controller Identifier String	GET ./Managers/{id} Extract value of Name property	Y
Set Management Controller Identifier	PATCH ./Managers/{id}	Y

IPMI Command	Redfish Equivalent	In OCP mockup
String	Request contains value of Name property	
Get Asset Tag	GET ./Systems/{id} Extract AssetTag	Y
Set Asset Tag	PATCH ./Systems/{id} Request contains AssetTag value	Y
Get Device ID	Not Supported. See section 5.2	N
Get System GUID	GET ./Systems/{id}#/UUID	Y
Set & Get System Boot Options	Supported. See section 5.3	Y
Get DCMI Sensor Info	GET ./TelemetryService/MetricDefinitions/{id}, where ID is the name of the sensor. Supported. See section 5.4	Y
Get Sensor Reading	GET <path to resource containing the sensor reading> Extract the value of the sensor reading See section 5.4	Y
Get SEL info	GET ./Systems/{id}/LogServices/SEL	Y
Get SEL Entry	GET ./Systems/{id}/LogServices/SEL/Entries/{id}	Y
Clear SEL	POST ./Managers/{id}/LogServices/SEL/Actions/LogService.Reset	Y
Get Power Reading	GET ./Chassis/{id}/Power Extract PowerControl/PowerConsumedWatts	Y
Get Temperature Readings	GET ./Chassis/{id}/Thermal Extract Temperatures	Y
Get LAN Configuration Parameters	GET ./Managers/{id}/EthernetInterfaces/{id}	Y
Set LAN Configuration Parameters	POST ./Managers/{id}/EthernetInterfaces/{id}/SD	Y
Set & Get Channel Access	Not supported. See section 5.5	N
Get User Access	GET ./AccountService/Accounts/{id} Extract the RoleId property	Y
Set User Access	PATCH ./AccountService/Accounts/{id} Request contains value of RoleId property	Y
Get User Name	GET ./AccountService/Accounts/{id} Extract the UserName property	Y
Set User Name	GET ./AccountService/Accounts/{id} Request contains value of UserName property	Y
Set User Password	GET ./AccountService/{id} Request contains the value of the Password property	Y
Set & Get User Payload Access	Not supported. See section 5.6	N
Get Chassis Capabilities	Not supported. See section 5.7	N
Get Chassis Status	GET ./Chassis/{id}#/State/Status	Y
Chassis Control	POST ./Systems/1/Actions/ComputerSystem.Reset Supported. See section 5.8	Y

IPMI Command	Redfish Equivalent	In OCP mockup
Chassis Identify	GET ./Chassis/{id}#/AssetTag	Y
Get ACPI Power State	GET ./Chassis/{id}#/ACPIPowerState	N

5.2 Device ID

IPMI supports a device ID to uniquely specify each device.

Redfish does not support the notion of Device ID.

5.3 Boot Property Structure

The System resource contains a boot property contains properties to control the booting of the system. The BootSourceOverrideTarget is a Redfish annotation which may be present to provide the client with the values that the implementation supports.

```
    "Boot": {
       "BootSourceOverrideEnabled": "Once",
       "BootSourceOverrideMode": "UEFI",
       "BootSourceOverrideTarget": "Pxe",
       "UefiTargetBootSourceOverride": "uefi device path"
       "BootSourceOverrideTarget@Redfish.AllowableValues": [
           "None",
           "Pxe",
           "Floppy",
           "Cd",
           "Usb",
           "Hdd",
           "BiosSetup",
           "Utilities",
           "Diags",
           "UefiTarget",
           "SDCard",
           "UefiHttp"
        ]
    }
```

5.4 Sensors

Regarding sensors, the OCP Specification refers to the DCMI. The DCMI spec specifies three sensors:

- Inlet Temperature
- Baseboard Temperature
- CPU (Processor) Temperature

In Redfish, these temperature metrics are contained in the Thermal resource. Within the resource, the Temperatures JSON object contains the temperature metrics associated with the Chassis named "Ch_1". Each temperature is identified by a "Name" property.

The following values of Name could be supported:

- "Inlet Temp", in which PhysicalContext="Intake"
- "Board Temp", in which PhysicalContext="SystemBoard"
- "CPU{n} Temp", where "{n}" is a unique integer value, in which PhysicalContext="CPU"

```
  "@odata.id": "/redfish/v1/Chassis/Ch_1/Thermal",

  "Temperatures": [
        {
            "@odata.id": "/redfish/v1/Chassis/1/Thermal#/Temperatures/0",
            "MemberId": "0",
            "Name": "Inlet Temp",
            "Status": { "State": "Enabled", "Health": "OK" },
            "ReadingCelsius": 25,
            "PhysicalContext": "Intake",
             . . .
        },
        {
            "@odata.id": "/redfish/v1/Chassis/1/Thermal#/Temperatures/1",
            "MemberId": "1",
            "Name": "Board Temp",
            "Status": { "State": "Enabled", "Health": "OK" },
            "ReadingCelsius": 35,
            "PhysicalContext": "SystemBoard"
             . . .
        },
        {
            "@odata.id": "/redfish/v1/Chassis/1/Thermal#/Temperatures/2",
            "MemberId": "2",
            "Name": "CPU1 Temp",
            "Status": { "State": "Enabled", "Health": "OK" },
            "ReadingCelsius": 45,
            "PhysicalContext": "CPU",
             . . .
        },
        {
            "@odata.id": "/redfish/v1/Chassis/1/Thermal#/Temperatures/3",
            "MemberId": "3",
            "Name": "CPU2 Temp",
            "Status": { "State": "Enabled", "Health": "OK" },
            "ReadingCelsius": 46,
            "PhysicalContext": "CPU",
             . . .
        }
```

Table 3 - Need a name here

DCMI Command	Redfish Equivalent	In OCP Profile
Get Inlet Temperature	GET ./Chassis/{id}/Thermal In the Temperature array property, find the Temperature, whose Name property is "Inlet Temp". From the Temperature so found, extract the value of ReadingCelsius	Y
Get Baseboard Temperature	GET ./Chassis/{id}/Thermal In the Temperature array property, find the Temperature, whose Name property is "Board Temp". From the Temperature so found, extract the value of ReadingCelsius	Y
Get CPU (Processor) Temperature	GET ./Chassis/{id}/Thermal In the Temperature array property, find the Temperature, whose Name property is "CPU{n} Temp", where "{n}" is a unique integer value From the Temperature so found, extract the value of ReadingCelsius	Y

The sensor section of DCMI specification also lists three event sensors.

- A power off event
- A power threshold event
- A thermal limit event

In Redfish, these power metrics are contained in the Power resource. Within the resource, the PowerControl JSON object contains the properties associated with the Chassis named "Ch_1". Each power control is identified by a "Name" property.

The following values of Name could be supported:

- "System Power Control", in which PhysicalContext="SystemBoard"

```
    "@odata.id": "/redfish/v1/Chassis/Ch_1/Power",
    "PowerControl": [
        {
            "@odata.id": "/redfish/v1/Chassis/1/Power#/PowerControl/0",
            "MemberId": "0",
            "Name": "System Power Control",
            "PowerConsumedWatts": 224,
            "PowerCapacityWatts": 600,
            "PowerLimit": {
                "LimitInWatts": 450,
                "LimitException": "LogEventOnly",
                "CorrectionInMs": 1000
            },
            "Status": { "State": "Enabled", "Health": "OK" },
        }
    ]
```

Redfish supports a subscribe/publish event model.

- For power off events

- The client would subscribes to a lifecycle events (EventType=" StatusChange")
- Upon receiving an event, inspect the Event Message and determine whether the MessageID property has the value of TBD

- For power threshold events, the client would TBD
- For thermal limit events, the client would TBD

5.5 Channel Access

IPMI supports a channel mechanism for routing messages to IPMB, and routing messages to different platform internal media. Channels are a specific capability of IPMI.

Redfish does not support channels.

5.6 User Payload Access

IPMI supports a payload mechanism to carry data besides IPMI payloads and Serial-over-LAN payloads over the IPMI protocol.

Redfish specifies HTTP/HTTPS. Redfish does not specify a special payload mechanisms. For Serial-over-LAN, the SSH or Telnet protocol could be supported.

5.7 Chassis Capabilities

IPMI supports chassis capabilities to return information about the chassis management which are present on the IPMB and how to access those functions. IPMB is a specific capability of IPMI.

Redfish does not support chassis capabilities.

5.8 Chassis Control

Redfish defines Chassis as the 'sheet metal' container with power and cooling domain. The generic definition of Chassis allows chassis to define any container.

Redfish encapsulates the other state actions besides power state actions in the ComputerSystem resource. The ComputerSystem resource defines the Action property.

The fragment defines an action names ComputerSystem.Reset, which the Redfish client can perform by POST'ing to the URI specified in the "target" property. The POST request should include a JSON file with the parameter(s) specified in the ResetActionInfo resource.

```
   "Actions": {
        "#ComputerSystem.Reset": {
           "target": "/redfish/v1/Systems/1/Actions/ComputerSystem.Reset",
           "@Redfish.ActionInfo": "/redfish/v1/Systems/1/ResetActionInfo"
        }
    }
```

Below is the contents of the ResetActionInfo resource. There is only one parameter specified which is required. The property name is "ResetType" and its value is a string. The structure also shows that allowable values that can be used as values.

```
{
    "@Redfish.Copyright": "...",
    "@odata.context": "/redfish/v1/$metadata#ActionInfo.ActionInfo",
    "@odata.id": "/redfish/v1/Systems/1/ResetActionInfo",
    "@odata.type": "#ActionInfo.v1_0_0.ActionInfo",
    "Parameters": [{
        "Name": "ResetType",
        "Required": true,
        "DataType": "String",
        "AllowableValues": [
            "On",
            "ForceOff",
            "GracefulShutdown",
            "GracefulRestart",
            "ForceRestart",
            "Nmi",
            "ForceOn",
            "PushPowerButton"
        ]
    }]
}
```

ANNEX A
(informative)

Change log

Version	Date	Description
0.2.2b	2017-05-30	Work in Progress