Redfish Host Interface: UEFI and OS Implications

Spring 2019 UEFI Plugfest - April 8-12, 2019
Mike Rothman, John Leung (Intel)
Samer El-Haj-Mahmoud (Lenovo)
Agenda

• Introduction
• New UEFI Redfish Interfaces
• Redfish Host Interface
• OS Support / Demo
• Summary and Q&A
What is Redfish™?

- A DMTF industry standard\(^1\)
- RESTful interface for managing IT Infrastructure
- Built on modern tool-chain (HTTPs/TLS, REST, JSON, OData)
- Schema-backed, human readable JSON output (including json-schema, OpenAPI)

\(^1\)dmf.org, redfish.dmtf.org
New Redfish UEFI Interfaces

- EFI REST EX Protocol
- EFI REST EX Servicing Binding Protocol
- EFI Redfish Discover Protocol
- EFI REST JSON Structure Protocol
- REST Style format in HII Question and Formset
UEFI Redfish Roadmap

<table>
<thead>
<tr>
<th>Before</th>
<th>Q4’18</th>
<th>Q1’19</th>
<th>Q2’19</th>
<th>Q3’19</th>
<th>Q4’19</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oct</td>
<td>Nov</td>
<td>Dec</td>
<td>Jan</td>
<td>Feb</td>
<td>Mar</td>
<td>Apr</td>
</tr>
</tbody>
</table>

- **Arch & POC**
- **Development**
- **Open Source**
- **EDK2 Staging**
- **Production**
- **EDK2 Master**
- **Community Co-engineering**

www.uefi.org
Open Source Feature Scope

- Support of EFI REST EX (Service Binding) Protocol
- Support of REST Style format in HII Question and Form set
- **DSP0266** (Redfish API), **DSP0270** (Redfish Host Interface) compliance
- UEFI Redfish Configuration Framework
  - Redfish ConfigHandler Protocol
  - Redfish Credential Protocol
- Redfish schema support (Bios, AttributeRegistry, ComputerSystem, BootOption)
- Sample Redfish Configurable Items
  - iSCSI Keywords
  - BootOrder/BootNext variables
UEFI Redfish Open Source code

• EDK Staging area
  – (will post once UEFI 2.8 published)
  – https://github.com/tianocore/edk2-staging
  – New “UEFI_Redfish” branch

• Call for feedback and contributions
  – EDK2 community and e-mail lists
    https://github.com/tianocore/tianocore.github.io/wiki/Mailing-Lists
Redfish Host Interface

- DMTF Host Interface Specification - DSP0270
  - “In-band” access to the Redfish service from UEFI/Host OS
  - Replacement for KCS/BT
  - Version 1.0.1 (Dec’17) and work-in-progress (~May'19)

- TCP/IP Based
  - Redfish HTTPs requests & responses over a TCP/IP network connection between Host/client and Manager/service.
  - Over any physical or logical interconnect that can route TCP/IP
Finding the Host Interface

Process
• The host discovers the supported Redfish manager interfaces
  – Accesses SMBIOS Type 42 structure for information on the BMC's Redfish Host Interface
  – Obtains information on the IP-based protocol needed to establish a connection
• The host initializes the host-side driver stack

Implementation
• OSes implement methods to
  – Find the Redfish Host Interface
  – Advertise to user space/applications
• This is the equivalent of Linux /dev/ipmiN or /dev/ipmidev/N for IPMI
  – Using KCS/BT information from SMBIOS Type 38 or ACPI SPMI table
### SMBIOS Specification: Table 42

<table>
<thead>
<tr>
<th>Offset</th>
<th>Name</th>
<th>Length</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>Type</td>
<td>BYTE</td>
<td>42</td>
<td>Management Controller Host Interface structure indicator</td>
</tr>
<tr>
<td>01h</td>
<td>Length</td>
<td>BYTE</td>
<td>Varies</td>
<td>Length of the structure, a minimum of 09h</td>
</tr>
<tr>
<td>02h</td>
<td>Handle</td>
<td>WORD</td>
<td>Varies</td>
<td></td>
</tr>
<tr>
<td>04h</td>
<td>Interface Type</td>
<td>BYTE</td>
<td>Varies</td>
<td>Management Controller Interface Type. 40h (Network Host Interface)</td>
</tr>
<tr>
<td>05h</td>
<td>Interface Specific Data Length (n)</td>
<td>BYTE</td>
<td>Varies</td>
<td>Interface-specific Data as specified by the Interface type</td>
</tr>
<tr>
<td>06h</td>
<td>Interface Specific Data</td>
<td>n BYTES</td>
<td>Varies</td>
<td>Defined by Interface Type</td>
</tr>
<tr>
<td>06h+n</td>
<td>Protocol count</td>
<td>BYTE</td>
<td>Varies</td>
<td>Number of protocols defined for the Host Interface (typically 1)</td>
</tr>
<tr>
<td>07h+n</td>
<td>Protocol Records</td>
<td>m Bytes</td>
<td>Varies</td>
<td>A Protocol Record for each protocol supported</td>
</tr>
</tbody>
</table>

![Diagram](image)
### Table 42: Interface Specific Data

<table>
<thead>
<tr>
<th>Offset</th>
<th>Name</th>
<th>Length</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Device Types supported</td>
<td>BYTE</td>
<td>Enum</td>
<td>Bits for USB, PCIe, USB v2, PCIe v2</td>
</tr>
<tr>
<td>X+1</td>
<td>Device Descriptors</td>
<td>n-1 Bytes</td>
<td>Varies</td>
<td>Device descriptor data formatted based on Device Type</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type value</th>
<th>Device Type Name</th>
<th>Length</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>02h</td>
<td>USB Network Interface</td>
<td>Varies</td>
<td>Varies</td>
<td>Device Descriptors for USB Device Type</td>
</tr>
<tr>
<td>03h</td>
<td>PCIe Network Interface</td>
<td>8 bytes</td>
<td>Varies</td>
<td>Device Descriptors for PCIe Device Type</td>
</tr>
<tr>
<td>04h</td>
<td>USB Network Interface v2</td>
<td>Varies</td>
<td>Varies</td>
<td>Device Descriptors for USB Device Type v2</td>
</tr>
<tr>
<td>05h</td>
<td>PCIe Network Interface v2</td>
<td>Varies</td>
<td>Varies</td>
<td>Device Descriptors for PCIe Device Type v2</td>
</tr>
<tr>
<td>80h-FFh</td>
<td>OEM</td>
<td>Varies</td>
<td>Varies</td>
<td>Device Descriptors for OEM Device Type</td>
</tr>
</tbody>
</table>

Note - USB and PCIe device types supported
## Table 42: Protocol Records data format

<table>
<thead>
<tr>
<th>Offset</th>
<th>Name</th>
<th>Length</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Protocol Identifier</td>
<td>BYTE</td>
<td>Varies</td>
<td>Protocol identifier: &quot;Redfish over IP&quot; = 04h</td>
</tr>
<tr>
<td>X+1</td>
<td>Length</td>
<td>BYTE</td>
<td>Varies</td>
<td>Length of protocol specific data</td>
</tr>
<tr>
<td>X+2</td>
<td>Protocol specific record data</td>
<td>p Bytes</td>
<td>Varies</td>
<td>Defined by protocol.</td>
</tr>
<tr>
<td></td>
<td>Service UUID</td>
<td>16 BYTEs</td>
<td>Varies</td>
<td>Same as Redfish Service UUID in Redfish Service Root resource; set to all 0s if the UUID is not supported or unknown.</td>
</tr>
<tr>
<td>X+16</td>
<td>Host IP Assignment Type</td>
<td>BYTE</td>
<td>Enum</td>
<td>Unknown, Static, DHCP, AutoConfigure, or HostSelected</td>
</tr>
<tr>
<td>X+17</td>
<td>Host IP Address Format</td>
<td>BYTE</td>
<td>Enum</td>
<td>Unknown, IPv4, or IPv6</td>
</tr>
<tr>
<td>X+18</td>
<td>Host IP Address</td>
<td>16 BYTEs</td>
<td>Varies</td>
<td>Used for Static and AutoConfigure.</td>
</tr>
<tr>
<td>X+34</td>
<td>Host IP Mask</td>
<td>16 BYTEs</td>
<td>Varies</td>
<td>Used for Static and AutoConfigure.</td>
</tr>
<tr>
<td>X+50</td>
<td>Redfish Service IP Discovery Type</td>
<td>BYTE</td>
<td>Enum</td>
<td>Unknown, Static, DHCP, AutoConfigure, or HostSelected</td>
</tr>
<tr>
<td>X+51</td>
<td>Redfish Service IP Address Format</td>
<td>BYTE</td>
<td>Enum</td>
<td>Unknown, IPv4, or IPv6</td>
</tr>
<tr>
<td>X+52</td>
<td>Redfish Service IP Address</td>
<td>16 BYTEs</td>
<td>Varies</td>
<td>Used for Static and AutoConfigure.</td>
</tr>
<tr>
<td>X+68</td>
<td>Redfish Service IP Mask</td>
<td>16 BYTEs</td>
<td>Varies</td>
<td>Used for Static and AutoConfigure.</td>
</tr>
<tr>
<td>X+84</td>
<td>Redfish Service IP Port</td>
<td>WORD</td>
<td>Varies</td>
<td>Used for Static and AutoConfigure.</td>
</tr>
<tr>
<td>X+86</td>
<td>Redfish Service VLAN ID</td>
<td>DWORD</td>
<td>Varies</td>
<td>Used for Static and AutoConfigure.</td>
</tr>
<tr>
<td>X+90</td>
<td>Redfish Service Hostname Length</td>
<td>BYTE</td>
<td>Varies</td>
<td>The length in bytes of the &quot;Redfish Service Hostname&quot; field</td>
</tr>
<tr>
<td>X+91</td>
<td>Redfish Service Hostname</td>
<td>varies</td>
<td>Varies</td>
<td>Hostname of Redfish Service</td>
</tr>
</tbody>
</table>
Redfish Host Interface Security

• Authentication, encryption, and authorization equivalent to the out-of-band Redfish API is supported
  – HTTPs/TLS, access restricted to authenticated BMC users, with proper privilege

• Implementations may also support AuthNone or un-encrypted connections when passing credentials
  – Should not be configured by default

• Implementations may also support Temporary BMC Credentials for OS root/admin access
  – Provisioned in UEFI Variables during system boot
  – OSes should read the UEFI variables early to retrieve the credentials, then delete - Not implemented in OSes yet
  – DMTF looking for alternative designs. Soliciting ideas at: https://redfishforum.com

www.uefi.org
dmidecode – ver 3.2+

- Decode SMBIOS Type 42 – Redfish Host Interface
- Support added by Neil Horman (Red Hat)
- Used by redfish-finder
- Sample output on Lenovo ThinkSystem SR650

```
[root@localhost ~]# dmidecode -t 42
# dmidecode 3.2
Getting SMBIOS data from sysfs.
SMBIOS 3.2.1 present.
# SMBIOS implementations newer than version 3.2.0 are not
# fully supported by this version of dmidecode.

Handle 0x2E30, DMI type 42, 169 bytes
Management Controller Host Interface
  Host Interface Type: Network
  Device Type: USB
    idVendor: 0x04b3
    idProduct: 0x4010
    Protocol ID: 04 (Redfish over IP)
    Service UUID: 6b6d716e-1eae-e711-a84e-9ce71daac05e
    Host IP Assignment Type: Static
    Host IP Address Format: IPv4
    IPv4 Address: 169.254.95.120
    IPv4 Mask: 255.255.0.0
    Redfish Service IP Discovery Type: Static
    Redfish Service IP Address Format: IPv4
    IPv4 Redfish Service Address: 169.254.95.118
    IPv4 Redfish Service Mask: 255.255.0.0
    Redfish Service Port: 443
    Redfish Service Vlan: 0
    Redfish Service Hostname: samer-sr650
```
redfish-finder

- Developed by Neil Horman: [https://github.com/nhorman/redfish-finder](https://github.com/nhorman/redfish-finder)
- Available with Fedora 30+. Coming to future Linux distros
- Parses the SMBIOS Type 42 data for Redfish access, and translates to an OS interface name
- Uses **NetworkManager** to configure the network interface with the appropriate IP settings
- Adds an entry to `/etc/hosts` mapping the name `redfish-localhost` to the discovered Redfish service address.
- Applications wishing to use the local redfish service can then point to the canonical url: [https://redfish-localhost/redfish/v1](https://redfish-localhost/redfish/v1)
redfish-finder demo
On Lenovo ThinkSystem SR650

[root@localhost ~]# dnf install
https://dl.fedoraproject.org/pub/fedora/linux//development/rawhide/Everything/x86_64/os/Packages/r/redfish-finder-0.3-1.fc31.noarch.rpm

[root@localhost ~]# redfish-finder
redfish-finder: Getting dmidecode info
redfish-finder: Building NetworkManager connection info
redfish-finder: Obtaining OS config info
redfish-finder: Converting SMBIOS Host Config to NetworkManager Connection info
redfish-finder: Applying NetworkManager connection configuration changes
Connection 'enp0s20f0u1u6' successfully deactivated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/2)
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/3)
redfish-finder: Adding redfish host info to OS config
redfish-finder: Done, BMC is now reachable via hostname redfish-localhost

[root@localhost ~]# cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4 ::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
169.254.95.118 redfish-localhost samer-sr650
redfish-finder demo

On Lenovo ThinkSystem SR650

[root@localhost ~]# curl -k https://redfish-localhost/redfish/v1/
{
  "Id":"RootService",
  "@odata.type":"#ServiceRoot.v1_2_0.ServiceRoot",
  "@odata.id":"/redfish/v1/",
  "@odata.context":"/redfish/v1/$metadata#ServiceRoot.ServiceRoot",
  "@odata.etag":"W/""5a216892b5be32e7faccd0a6f16007d0"",
  "Name":"Root Service",
  "RedfishVersion":"1.0.2",
  "Description":"This resource is used to represent a service root for a Redfish implementation.",
  "UUID":"E7C98E86-7D03-461F-9519-CB5FE0F45A63",
  "Chassis":{"@odata.id":"/redfish/v1/Chassis/"},
  "Managers":{"@odata.id":"/redfish/v1/Managers/"},
  "Systems":{"@odata.id":"/redfish/v1/Systems/"},
  "JsonSchemas":{"@odata.id":"/redfish/v1/JsonSchemas/"},
  "Registries":{"@odata.id":"/redfish/v1/Registries/"},
  "Tasks":{"@odata.id":"/redfish/v1/Tasks/"},
  "SessionService":{"@odata.id":"/redfish/v1/SessionService/"},
  "EventService":{"@odata.id":"/redfish/v1/EventService/"},
  "AccountService":{"@odata.id":"/redfish/v1/AccountService/"},
  "UpdateService":{"@odata.id":"/redfish/v1/UpdateService/"}
  "Links":{"Sessions":{"@odata.id":"/redfish/v1/SessionService/Sessions/"}},
  "Oem":{"Lenovo":{"FirmwareServices":{"@odata.id":"/redfish/v1/Oem/Lenovo/FirmwareServices/"}}},
}
fwupd and LVFS

- fwupd\(^1\) and LVFS\(^1\): Linux Vendor Firmware Service
- Work by Richard Hughes
- Popular on Client devices
  - Using UEFI Capsules and ESRT
  - Not typically supported on servers
- Added **Redfish Plugin**:  
  - [https://github.com/hughsie/fwupd/tree/master/plugins/redfish](https://github.com/hughsie/fwupd/tree/master/plugins/redfish)
  - Enables FW Update “in-band” on Redfish conformant servers
  - Uses SMBIOS Type 42 to find the Redfish Host Interface
  - Does not rely on redfish-finder (yet)
OpenBMC Redfish Support

- OpenBMC is a Linux Foundation project\(^1\)
- OpenBMC 2.6 supports Redfish\(^2\) (Feb 2019)
  - Expect bi-annual releases
- Redfish Host Interface is on the “request for enhancement” list
  - This is the host replacement of KCS/BT

\(^1\) openbmc.org
\(^2\) github.com/openbmc/openbmc/releases/tag/2.6.0

www.uefi.org
Call To Action

- **Implement Redfish Host Interface in your firmware**
  - BMC firmware – OpenBMC, OEMs, BMC vendors, ISVs
  - UEFI System firmware – EDK2 open source, OEMs, IBVs

- **Implement Redfish Host Interface in your OS**
  - OSVs, ISVs, open source community

- **Use Redfish Host Interface in applications**
  - From OS kernel or user space
  - User tools/scripts running in the OS

- **Provide feedback to DMTF!**
  - Your DMTF Redfish Forum member company representatives
  - Or open users forum: [http://www.redfishforum.com](http://www.redfishforum.com)

www.uefi.org
Questions

- **Redfish User Forum**
  - User forum for questions, suggestions and discussion of all Redfish topics
  - [http://www.redfishforum.com](http://www.redfishforum.com)

- **Redfish Developer Portal**
  - Redfish Interactive Resource Explorer
  - Educational material, Hosted Schema files, documentation & other links
  - [http://redfish.dmtf.org](http://redfish.dmtf.org)

- **Redfish Standards page**
  - Schemas, Specs, Mockups, White Papers, FAQ, Educational Material & more
  - [http://dmtf.org/redfish](http://dmtf.org/redfish)

- **DMTF Redfish Forum**
  - Companies involved, Upcoming Schedules & Future work, Charter
  - Join the DMTF to get involved in future work
  - [http://www.dmtf.org/standards/spmf](http://www.dmtf.org/standards/spmf)
Thanks for attending the 2019 Spring UEFI Plugfest

For more information on UEFI Forum and UEFI Specifications, visit http://www.uefi.org

presented by